

**Predictie van
interconnectie-eigenschappen
van digitale schakelingen
voor de exploratie van
ontwerpkeuzes en technologieën**

*Prediction of interconnect properties
for digital circuit design and technology exploration*

Joni Dambre

Promotoren: prof. dr. ir. D. Stroobandt
prof. dr. ir. J. Van Campenhout

Proefschrift ingediend tot het behalen van de graad van
Doctor in de Toegepaste Wetenschappen:
Computerwetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. J. Van Campenhout
Academiejaar 2002-2003



*Ter herinnering aan
Alena Raes,
wie ze was
en wat ze deed*

Dankwoord

De Gentse universiteit speelde al heel vroeg een rol in mijn leven. Op heel jonge leeftijd maakte ik op de vierde verdieping van de Ledeganckstraat de gangen en bureaus onveilig in het labo waar mijn moeder werkte. Mijn lagere schooltijd spendeerde ik aan het toenmalig proefschooltje van de faculteit pedagogiek. 's Avonds keek ik soms over de schouders van studenten die, onder toezicht van mijn moeder, dissecties en andere practicumproeven uitvoerden. Het Technicum verkende ik voor het eerst toen ik op de middelbare school zat, toen mijn moeder foto's kwam nemen in het labo voor elektronenmicroscopie.

Toen ik mijn studies burgerlijk ingenieur begon voelde de omgeving al bijna als een tweede thuis. Met de inhoud had ik aanvankelijk meer moeite en even was opgeven niet veraf. Pas na een uiteenzetting van professor Vanwormhoudt over de richting elektrotechniek voelde ik dat het schoentje "ingenieur" mij toch paste. Zonder zijn invloed, zowat 10 jaar geleden, was dit doctoraat er wellicht nooit gekomen. Ik ben hem dan ook dankbaar voor de begeestering die hij ons toen meegaf en omdat hij nu in mijn jury wilde zetelen.

Met zo'n van universiteit doordrenkte jeugd lijkt het niet meer dan natuurlijk dat ik ook na mijn studies aan deze universiteit ben "blijven hangen". Toch was mijn verbazing wel erg groot toen professor Jan Van Campenhout, mijn promotor, meteen positief reageerde op mijn voorzichtige vragen over "aan de universiteit blijven". Nu, zeven jaar later, ben ik heel blij dat hij als voorwaarde stelde dat ik zou doctoreren. Dat ik dat in zijn onderzoeksgroep mocht doen beschouw ik als een buitenkans: hij gaf mij de inspiratie, maar vooral de vrijheid en het vertrouwen om, vaak kriskras, mijn eigen(zinnige) ideeën te volgen. Terzelfdertijd werkte ook zijn aandacht voor juistheid en grondigheid in onderzoek en rapportering aanstekelijk. Bedankt dus, Jan, voor alles wat je mij als onderzoeker hebt bijgebracht.

Wat de inhoudelijke kant van deze doctoraatsthesis betreft, ben ik vooral professor Dirk Stroobandt, mijn tweede promotor, en Herwig Van Marck erkentelijk voor het elan dat zij gaven aan het boeiende en nu ook bloeiende onderzoeksdomein rond SLIP-technieken. Misschien dankzij het feit dat we het vaak niet eens waren, vormden de discussies met Herwig en later met Peter steeds een vruchtbare bron van nieuwe ideeën. Met Dirk deel ik vooral het geruststellende gevoel dat we in dit onderzoeksdomein (en vaak ook daarbuiten) grotendeels dezelfde visie hebben. Bedankt ook, Dirk, voor een goede en vriendschappelijke samenwerking die we hopelijk nog een tijdje kunnen verderzetten.

Dr. Phillip Christie, some time ago your ideas were the spark that ignited Herwig's interest in interconnect prediction. Many years later your work got me interested in statistical mechanics. You were also the one to suggest that I perform a thorough evaluation of SLIP techniques, which resulted in an important part of this Ph.D. thesis. Since I consider you to be one of the most creative and innovative people in the SLIP research domain, I am very grateful to you for being a member of my jury.

Professor Patrick Groeneveld van de TU Eindhoven wil ik bedanken voor zijn onmiddellijke bereidheid om in mijn jury te zetelen. Zijn opmerkingen, gestoeld op een ruime praktijkervaring, waren zeer waardevol en ik hoop dat we in de toekomst samen kunnen werken om de kloof tussen theorie en praktijk zo ver mogelijk te dichten.

Mijn dank gaat ook uit naar professor Francky Catthoor van IMEC voor de grondigheid waarmee hij mijn werk heeft gelezen. Zijn visie op het ontwikkelen van ontwerpmethodieken gaf mij enige tijd geleden nieuwe hoop over de toekomst van SLIP-technieken. Zijn opmerkingen over ontwerpaspecten waarmee ik nog niet zo vertrouwd ben, geheugenexploratie en vermogenverbruik, waren zeer verrijkend en openen verschillende pistes voor verder onderzoek.

Hoewel daar in dit werk niet veel van terug te vinden is, situeerde mijn onderzoek zich oorspronkelijk in het domein van optische interconnecties. Ik werd opgenomen in een zeer actief onderzoeksnetwerk rond dit thema. De samenwerking in opeenvolgende projecten was niet alleen zeer leerrijk, maar heeft ook gezorgd voor veel goede contacten. Twee leden van dit netwerk, professoren Roel Baets en Hugo Thienpont, wil ik in het bijzonder bedanken voor hun bereidheid om in mijn jury te zetelen, ondanks het feit dat optische interconnecties niet expliciet in mijn thesis aanwezig waren.

Minstens even belangrijk als het onderzoek zelf was de sfeer in de

onderzoeksgroep. Een doctoraatsonderzoek wordt gekenmerkt door enkele hoogtepunten en veel tegenslagen. In die omstandigheden is een menselijke en vriendschappelijke omgang met promotor en collega's essentieel. Ik wil dan ook Jan en de collega's van het eerste uur bedanken voor het luisterend oor, de steun en de inspiratie. Bedankt Marnik, Herwig, Henk, Dirk, Wim, Peter en vooral Jo, met wie ik jarenlang lief, leed en een bureau ruimte deelde. Als nieuwe bureau-genoot was Michiel De Wilde steeds vriendelijk en behulpzaam. Een extra woordje van dank gaat uit naar Ronny, Marnik, Wim, Kristof en de beide Michiels voor de hulp bij de vele computerproblemen en -spoken die mijn pad kruisten, en ook naar Benjamin, Wim en Jan De Ceuster om veel van mijn onderwijstaken over te nemen zodat ik in alle rust kon schrijven. Natuurlijk wil ik ook de andere collega's niet vergeten: de rest van "PARIS-haut" (Hendrik, Harald, Mark en Erik) evenals Koen en de snel groeiende groep van "PARIS-bas".

Ook bedankt voor de steun die ik kreeg van de OAP-collega's. In het bijzonder moet ik hiervoor Rik Van de Walle bedanken, die mij met zijn gebruikelijke enthousiasme naar mijn eerste OAP-vergadering sleurde. En om het sociale weefsel van de werksfeer helemaal uit te spinnen, kijk ik ook met plezier terug op heel wat aangename lunch-gesprekken: Ludo, Pieter, Johan, Serge en Fons in mijn eerste jaren, de PARIS-collega's en Wilfried in het heden.

Op het thuisfront gaat mijn dank natuurlijk uit naar mijn ouders, voor de steun en voor de zeer gunstige start die ik in mijn leven kreeg. Tijdens mijn doctoraatsonderzoek kon mijn moeder mijn ups en downs steeds vanuit haar eigen ervaring relativeren. Haar wil ik ook bedanken voor de zichtbare en voelbare inspanning om mij te ontzien tijdens het schrijven van dit boek. Mijn vader volgde weliswaar van op een afstand, maar met niet minder trots de avonturen van zijn dochter. Ook mijn broer Bram wil ik bedanken om er te zijn. Hoewel onze violen niet steeds gelijk klonken wist ik dat ik altijd op hem kon rekenen.

Je veux remercier Dany pour m'accepter avec tant de chaleur et pour les conversations (longues et intéressantes) au téléphone. Et à Alexandre: je suis sûr que tu finiras tes études et que tu seras un très bon ingénieur.

Tenslotte wil ik Griet bedanken voor haar niet-aflatende steun gedurende bijna de hele periode van mijn onderzoek.

Joni Dambre
Gent, 7 juli 2003

Examencommissie

Voorzitter: prof. P. Kiekens, decaan
FTW, Universiteit Gent

Secretaris: prof. J. Debacker
FTW, Universiteit Gent

Leden: prof. dr. ir. R. Baets
INTEC, FTW, Universiteit Gent

prof. dr. ir. F. Catthoor
DESICS, IMEC, Leuven

dr. ir. P. Christie
CMOS Module Integration,
Philips Research Leuven

prof. dr. ir. P. Groeneveld
ICS, Faculteit elektrotechniek,
Technische Universiteit Eindhoven, NL

prof. dr. ir. D. Stroobandt
ELIS, FTW, Universiteit Gent

prof. dr. ir. H. Thienpont
TONA-TW, Vrije Universiteit Brussel

prof. dr. ir. J. Van Campenhout
ELIS, FTW, Universiteit Gent

prof. dr. ir. M. Vanwormhoudt
ELIS, FTW, Universiteit Gent

Nederlandstalige samenvatting

Inhoud

1	Inleiding	i
1.1	Het ontwerp van digitale schakelingen	i
1.2	Het stijgende belang van interconnecties	v
1.3	Optimalisatie van de interconnectiekost tijdens de verschillende ontwerpstappen	vi
1.4	A priori schatting van interconnectie-eigenschappen	vii
1.5	Belangrijkste bijdragen van dit doctoraatsonder- zoek	ix
2	Circuitgrafen en hun plaatsing	xii
2.1	Circuitgrafen	xii
2.2	De regel van Rent	xiii
2.3	De architectuur	xv
2.4	Het plaatsingsmodel	xv
2.5	Een woordje over de experimenten	xvii
3	Basistechnieken voor de voorspelling van interconnec- tielengtes	xviii
3.1	De schattingsmethode van Donath	xviii
3.2	De schattingsmethode van Davis	xxi
4	Het aantal pinnen van verschillende moduletypes	xxiv
4.1	Partitionerings-Rentkarakteristieken	xxiv
4.2	Geometrische Rentkarakteristieken	xxvii
4.3	Algemeen besluit uit deze analyse	xxix
5	Een meer nauwkeurige voorspelling van interconnectielengtes	xxix
5.1	Verbetering van de betrouwbaarheid	xxix
5.2	Verbetering van de nauwkeurigheid	xxx
5.3	Gevoeligheid aan de gebruikte Rentkarakteristiek	xxxii
6	Voorspelling van de minimale klokperiode	xxxiii
6.1	De klokperiode en het traagste pad	xxxiii
6.2	Een probabilistische benadering	xxxiii

6.3	Resultaten en besluit	xxxiv
7	Enkele ideeën voor verder onderzoek	xxxv
7.1	Verfijningen van en uitbreidingen aan de voorge- stelde modellen	xxxv
7.2	Verstevinging van de theoretische onderbouw . . .	xxxvi
8	Conclusies	xxxviii
8.1	Analyse	xxxviii
8.2	Schatting van interconnectie-eigenschappen . . .	xxxix

Nederlandstalige samenvatting

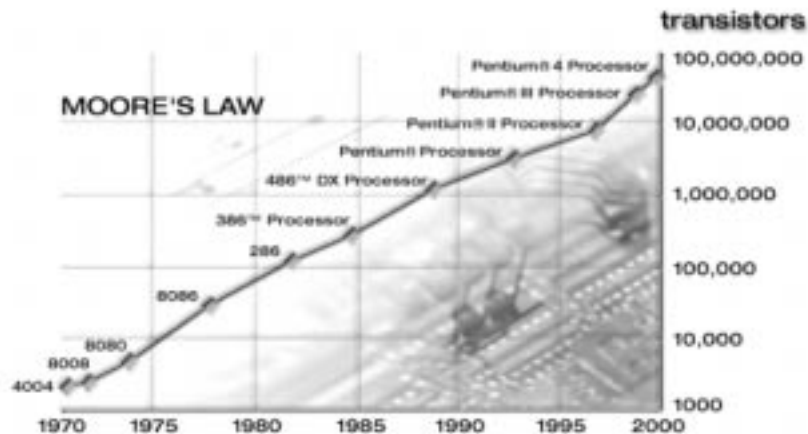
1 Inleiding

1.1 Het ontwerp van digitale schakelingen

Al geruime tijd verdubbelt het aantal transistors op een digitale chip ongeveer om de 18 maanden (wet van Moore, Figuur 2.1). Moderne chips bevatten miljoenen transistors en de integratiedichtheid neemt nog steeds toe. Het ontwerp van dergelijke chips kan enkel gebeuren dankzij een zeer systematische ontwerpmethodiek, waarvan de verschillende stappen voor een groot gedeelte worden uitgevoerd door software-pakketten (CAD of *Computer-Aided Design*).

Het ontwerp van digitale schakelingen begint bij de *specificaties*, een omschrijving van het gewenste gedrag van de schakeling en de randvoorwaarden waaraan de uiteindelijke implementatie moet voldoen. Deze randvoorwaarden kunnen bijvoorbeeld betrekking hebben op de minimale klokfrequentie, het maximale vermogenverbruik, de betrouwbaarheid of de productiekost.

Omdat het aantal mogelijke implementaties niet te bevatten is, gebeurt de exploratie van de verschillende oplossingsmogelijkheden (*de ontwerpruimte*) op een hiërarchische manier. Op elk hiërarchieniveau is slechts een gedeelte van de opbouw en het gedrag van het ontwerp zichtbaar. Binnen dat hiërarchieniveau wordt dan, op basis van schattingen voor de belangrijkste kostmetrieën (snelheid, vermogenverbruik, productiekost, enz.), gezocht naar de beste manier om het gewenste gedrag te bereiken. Bij de overgang naar een volgend hiërarchieniveau wordt meer detailinformatie zichtbaar gemaakt, en worden bijgevolg ook de gemaakte schattingen nauwkeuriger. De

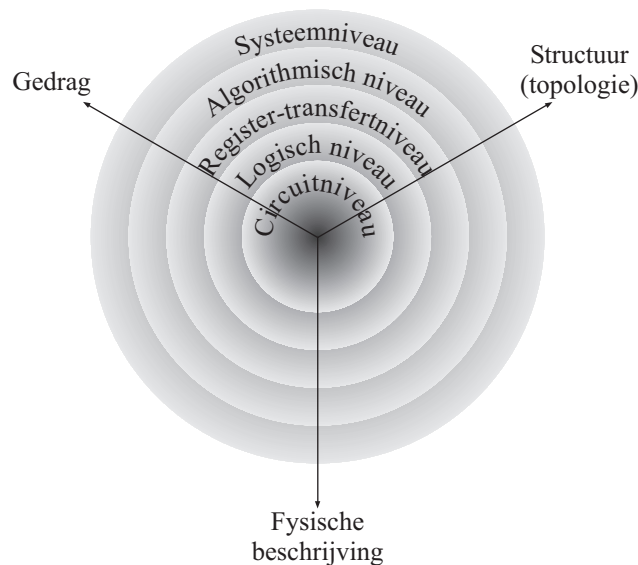


Figuur 2.1: Illustratie van de wet van Moore, afkomstig van <http://www.intel.com/>.

zoekruimte binnen het nieuwe hiërarchieniveau betreft echter enkel verfijningen van de oplossing die op het vorige niveau gevonden werd. Op die manier kan men de zoekruimte enorm beperken. In een efficiënte ontwerpmethodiek optimaliseert men de eigenschappen met de grootste impact op de uiteindelijke kost op een zo hoog mogelijk ontwerpniveau.

Een vaak gebruikte voorstelling van de ontwerpruimte is het Y-diagram van Gajski (Figuur 2.2). Daarin wordt voor elk ontwerpniveau een onderscheid gemaakt tussen verschillende manieren om het ontwerp te beschrijven: het gedrag, de structuur en de fysische implementatie. Het ontwerpproces kan in dit Y-diagram worden voorgesteld als een *ontwerptrajectorie*, die de hiërarchieniveaus van boven naar onder doorloopt. Het einddoel van het ontwerpproces is een fysische beschrijving op circuitniveau. Deze bevat alle informatie die nodig is om de chip te kunnen fabriceren. Een typisch overzicht van de belangrijkste stappen in het ontwerpproces wordt gegeven in Figuur 2.3. Hieronder volgt een korte beschrijving van deze stappen.

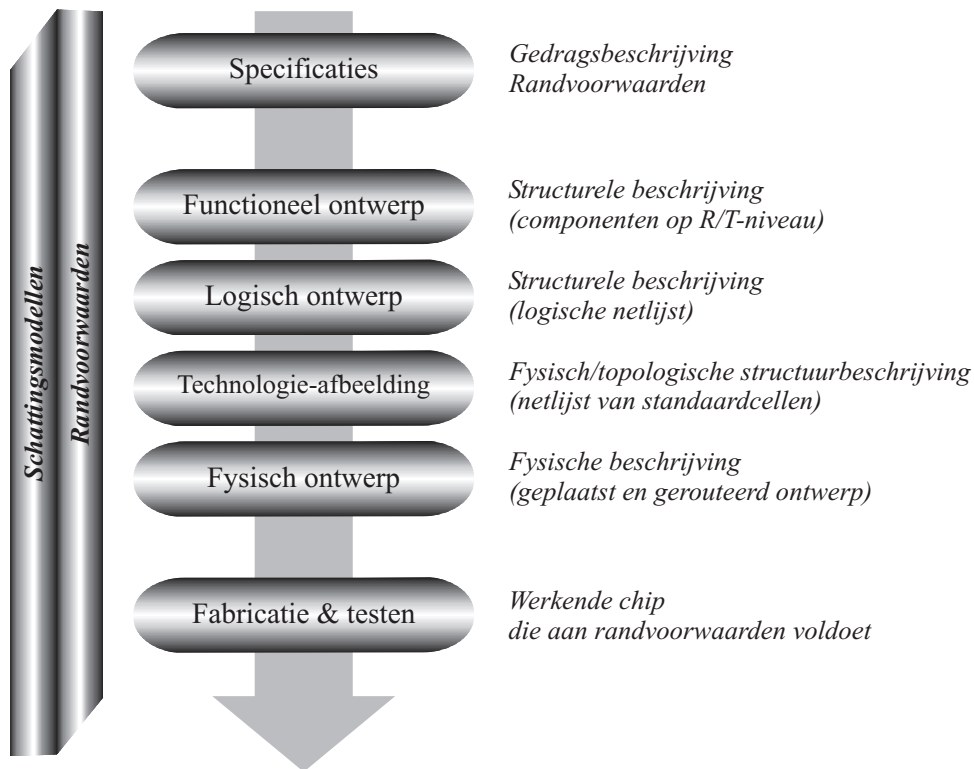
- De specificaties van het ontwerp worden eerst geformaliseerd tot een hoofdzakelijk gedragsmatige beschrijving op systeemniveau.
- De eerste stap na het formaliseren van de specificaties wordt vaak *functioneel ontwerp* genoemd. Deze stap bevat vaak verschillende



Figuur 2.2: Gajski's Y-diagram: een voorstelling van de ontwerpniveaus en verschillende beschrijvingen.

niveaus van gedragsverfijning, telkens gevolgd door een *synthese-stap*, waarbij wordt overgegaan van een gedragsbeschrijving naar een structuurbeschrijving. De deelblokken in de structuurbeschrijving worden dan bij de volgende stap verder ingevuld. Het eindresultaat van het functioneel ontwerp is meestal een structuurbeschrijving op register-transfertiiveau (R/T-niveau). Deze bestaat uit typische combinatorische bouwblokken zoals optellers en multiplexers, evenals sequentiële controle-automaten, registerbanken en geheugenblokken.

- De volgende stap is die van de *logische synthese*. Daarbij worden de R/T-bouwblokken verder uitgewerkt en geoptimaliseerd, eerst in hun gedragsrepresentatie (logische functies) en vervolgens in hun structurele representatie. Deze laatste bestaat uit geïnterconnecteerde elementaire logische poorten en flipflops en wordt *logische netlijst* genoemd.
- De meeste digitale chips zijn opgebouwd uit standaardcellen. Dit zijn elementaire bouwblokken die veelal reeds tot op fysisch niveau volledig ontworpen zijn. Hun beschrijvingen zijn beschikbaar via een bibliotheek. De logische netlijst moet dus worden



Figuur 2.3: De belangrijkste stappen in het ontwerpproces. De modellen voor het schatten van de ontwerpqualiteit worden, na de nodige abstractie, doorgegeven van de lagere naar de hogere ontwerp niveaus, terwijl de randvoorwaarden mee met het ontwerp van boven naar beneden propageren.

omgezet in een netlijst van geïnterconnecteerde standaardcellen, waarbij men beperkt is tot wat beschikbaar is in de bibliotheek. Deze stap wordt *technologie-afbeelding* (in het Engels *technology mapping*) genoemd. Het resultaat is een netlijst die zowel structurele elementen (de interconnectietopologie) als fysische elementen (de invulling van de cellen) bevat. In hedendaagse ontwerpen worden ook vaak grotere, vooraf gedefinieerde bouwblokken gebruikt: IP-blokken (IP = *intellectual property*) en kleine geheugenblokken.

- Om te komen tot een volledig fysische implementatie van de chip, dienen eerst alle componenten een plaats te krijgen op het chipoppervlak. Deze *plaatsing* dient zo te gebeuren dat de interconnecties zo kort mogelijk zijn. Vervolgens moet worden beslist

welk pad precies gevolgd moet worden door elke verbinding. Dit noemt men *routing*. Zowel plaatsing als routing gebeuren vaak in meerdere stappen, volgens het principe van hiërarchische verfijning. Aangezien hier volledig wordt overgegaan naar een fysische beschrijving, worden plaatsing en routing samen ook wel *fysisch ontwerp* genoemd.

Zoals reeds gezegd, worden in elk van deze ontwerpstappen schattingen gemaakt van verschillende kostmetriekeken zoals oppervlakte, vermogenverbruik, klokperiode en productiekost. Deze schattingen zijn gebaseerd op fysische modellen voor deze eigenschappen. Aangezien op hogere hiërarchieniveaus minder concrete informatie beschikbaar is, worden ze van lagere naar hogere ontwerpniveaus doorgegeven door ze te *abstraheren*. Dit gaat steeds gepaard met een verlies aan nauwkeurigheid. Om toch nog betrouwbare afwegingen te kunnen maken tussen verschillende implementatiekeuzes, is het belangrijk dat de opeenvolgende abstracties minstens de *correlatie* van de schattingen met de werkelijke kost zo goed mogelijk behouden.

Bij het doorlopen van de ontwerptrajectorie worden ook de opgelegde randvoorwaarden steeds verder verfijnd. Het komt vaak voor dat men op een bepaald ontwerpniveau ontdekt dat er toch verkeerde keuzes gemaakt zijn, waardoor de uiteindelijke randvoorwaarden niet meer kunnen gehaald worden. Op dat moment dient men terug te gaan naar een vorige stap om bij te sturen. Daardoor is in werkelijkheid de ontwerptrajectorie vaak veel minder lineair dan hierboven beschreven.

1.2 Het stijgende belang van interconnecties

In het verleden werden de meeste kostmetriekeken gedomineerd door het aantal functionele blokken en hun eigenschappen. Zij namen het grootste gedeelte van de oppervlakte in beslag en waren verantwoordelijk voor het grootste gedeelte van de vertragingen. Daarom zijn in traditionele ontwerpstechnieken de meeste afwegingen en optimalisaties afgestemd op het minimaliseren van het aantal poorten (oppervlakte, vermogenverbruik) of op het beperken van de *logische paddiepte* of het maximaal aantal poorten op een combinatorisch pad (klokperiode).

Tegenwoordig speelt de kost van interconnecties echter een steeds grotere rol. Voor sommige metriekeken worden interconnecties zelfs de dominante factor in de totale kost. Dit heeft voor een groot gedeelte te maken met het feit dat de eigenschappen van interconnecties en lo-

gische poorten een ander schalingsgedrag vertonen als gevolg van de toenemende miniaturisatie.

Als voorbeeld beschouwen we de evolutie van de vertragingen, veroorzaakt door poorten en interconnecties. De vertraging van interconnecties wordt voor een groot gedeelte bepaald door hun gedistribueerde weerstand r en capaciteit c per meter en door hun lengte. We kunnen de tijdsconstante van een verbinding schatten als

$$\tau \simeq RC = rcl^2.$$

Veronderstel nu dat alle afmetingen op de chip afnemen met een factor $S < 1$. In dit geval blijft de interconnectiecapaciteit c ongeveer gelijk, maar de weerstand neemt toe volgens $1/S^2$. Anderzijds blijft de totale chipoppervlakte ongeveer gelijk of neemt ze zelfs toe: het verkleinen van de poorten wordt benut om meer functionaliteit op één chip te kunnen plaatsen. De lengte van de langste verbindingen verandert dus ook nauwelijks, zodat de tijdsconstante van de langste verbindingen ongeveer toeneemt met een factor $1/S^2$. Voor de poorten is dit echter veel minder het geval: doordat de kanalen korter worden en de gate-oppervlakte kleiner, nemen zowel de uitgangsweerstand als de ingangscapaciteit van de poorten af. Het globale effect van dit schalingsgedrag en van een aantal meer ingewikkelde fenomenen die steeds belangrijker worden, is dat het aandeel van de verbindingen in de totale vertraging snel toeneemt.

Om de vertraging van verbindingen te beperken, worden de uitgangsbuffers van de poorten sterker gemaakt en worden lange verbindingen onderbroken door het invoegen van buffers. Dit gaat echter ten koste van de chipoppervlakte en het vermogenverbruik en kan dus niet onbeperkt worden toegepast.

Daarnaast zijn er bijvoorbeeld ook beperkingen aan de minimale afstand die er tussen de verbindingen op een chip moet zijn (ruis, overspraak, ...). Daardoor moet men steeds meer interconnectielagen aanbrengen om toch alle verbindingen te kunnen leggen.

1.3 Optimalisatie van de interconnectiekost tijdens de verschillende ontwerpstappen

Door het toenemend belang van verbindingen in de chipkost, worden steeds meer ontwerpstappen aangepast om hier rekening mee te houden. Het probleem is echter dat de lengte van de verbindingen, die zo

belangrijk is voor de verschillende kostmetrieke, pas bekend wordt tijdens fysisch ontwerp.

De eerste ontwerpstappen waarin men meer aandacht aan interconnecties ging besteden waren dan ook de fysische ontwerpstappen. Vroeger werd daarin reeds de totale verbindingslengte in rekening gebracht, omdat die invloed had op de chipoppervlakte en het aantal routeringslagen. Tegenwoordig wordt echter, o.m., de bijdrage van de verbindingen aan de minimale klokperiode veel sterker in rekening gebracht.

Door de onvoorspelbaarheid van de interconnectielengtes moet men in het ontwerpproces steeds vaker iteraties inbouwen. Wanneer men bijvoorbeeld teruggaat tot op het niveau van logische synthese en het volledige fysisch ontwerp opnieuw doet, is de kans groot dat men, door de oude problemen op te lossen, nieuwe problemen introduceert. Daarom tracht men de iteraties zo klein mogelijk te maken door zoveel mogelijk lokale wijzigingen aan te brengen. Daarbij wordt het gedeelte van de plaatsing en de routing dat reeds was uitgevoerd zo veel mogelijk behouden.

Toch blijven deze verbeteringen aan de ontwerpmethodologie beperkt tot fysisch ontwerp, waar reeds een groot gedeelte van de uiteindelijke interconnectiekost gekend is. Voor gedrags- en structurele optimalisaties vóór fysisch ontwerp wordt nog steeds in hoofdzaak rekening gehouden met de kost van logische poorten. Om toch reeds ruwe schattingen van de interconnectielengtes beschikbaar te maken voor deze ontwerpstappen, stellen sommige onderzoekers voor om reeds zo vroeg mogelijk in het ontwerpproces een eerste, zeer grove plaatsing uit te voeren (predictie door constructie).

1.4 A priori schatting van interconnectie-eigenschappen

Al de reeds vermelde technieken helpen om het ontwerpproces sneller tot een goed einde te brengen. Nochtans bieden ze geen fundamenteel antwoord op de vraag hoe men de interconnectiekost van bij het begin van het ontwerpproces in rekening kan brengen. Met andere woorden: welke eigenschappen van, bijvoorbeeld, een logische netlijst bepalen hoe lang de verbindingen zullen zijn na plaatsing en routing, en hoe kan men op basis van die eigenschappen deze verbindingslengtes schatten?

Deze vragen behoren tot het onderzoeksdomein van de *a priori*

schatting van interconnectie-eigenschappen (in het Engels: *System-Level Interconnect Prediction* of SLIP). Het oudste werk in deze context dateert van de late jaren 1960, toen de *regel van Rent* voor het eerst werd beschreven. Deze empirische regel vormt nu nog steeds de basis van de meeste SLIP-technieken.

De meest gebruikte modellen voor het schatten van de interconnectiekost gaan uit van een model voor de interconnectietopologie van de schakeling en een model voor de fysische en metrische eigenschappen van het implementatiesubstraat (ook wel de *architectuur* genoemd). Deze worden gecombineerd met een model voor het fysisch ontwerp of veronderstellingen omtrent het resultaat ervan. Wat men tracht te voorspellen zijn interconnectielengtes en eigenschappen die daarvan afgeleid kunnen worden, zoals vertraging en minimale klokperiode, chipoppervlakte en benodigd aantal routeringslagen. Om de exploratie van de ontwerpruimte op betrouwbare wijze te ondersteunen, moeten deze schattingen niet extreem nauwkeurig zijn. Ze moeten echter wel een goede correlatie vertonen met de achteraf gemeten waarden. Als ze bovendien ook relatief nauwkeurig zijn, is dit een extra pluspunt.

Eenmaal met behulp van dergelijke schattingen één of enkele goede oplossingen van het ontwerpvragestuk zijn geselecteerd, kunnen de reeds eerder vermelde technieken voor de optimalisatie van interconnectie-eigenschappen worden gebruikt om het ontwerp verder af te werken.

Aangezien schattingsmodellen aan nauwkeurigheid en betrouwbaarheid inboeten naarmate ze geabstraheerd worden, is het belangrijk om te starten van zo goed mogelijke schattingen van de interconnectielengtes. Deze kunnen dan worden gecombineerd met bestaande modellen voor, bijvoorbeeld, de interconnectievertraging.

Omdat het hier gaat om *a priori* schattingen, moeten we ons dus in eerste instantie concentreren op voorspellingen op basis van een netlijst van standaardcellen, logische poorten of (eventueel) bouwblokken op R/T-niveau. Hoewel sommige schattingstechnieken op dit niveau al geruime tijd bestaan, worden ze alsnog nauwelijks gebruikt voor de exploratie van de ontwerpruimte. De weinige toepassingen waarin ze echt zijn doorgebroken situeren zich op het gebied van de evaluatie en exploratie van nieuwe technologieën. Daarin tracht men in te schatten wat de voordelen zouden zijn van verschillende (toekomstige) technologische vernieuwingen. Een van de meest gekende voorbeelden hiervan is de *International Technology Roadmap for Semiconductors* (ITRS). In

de meeste van deze toepassingen streeft men geen schattingen na voor individuele schakelingen, maar voor *klassen* van schakelingen, waarvan men dan de *typische* eigenschappen in rekening probeert te brengen.

1.5 Belangrijkste bijdragen van dit doctoraatsonderzoek

De lange-termijndoelstelling waarin dit doctoraatsonderzoek kadert is het komen tot nauwkeurige voorspellingen van de interconnectie-eigenschappen van individuele schakelingen. Deze voorspellingen moeten significante kwaliteitsverschillen tussen verschillende oplossingen evenals tussen verschillende technologische keuzes weerspiegelen. Het onderscheidend vermogen van de bestaande technieken is hiervoor onvoldoende.

In de meest algemene zin omvat deze doelstelling een veelheid aan deelproblemen. De meest dringende hiervan zijn de slechte correlatie met meetresultaten voor individuele schakelingen, het correct in rekening brengen van verbindingen die meer dan twee deelblokken verbinden en het identificeren van lokale effecten die voor problemen kunnen zorgen tijdens de routeringsfase. Naar onze mening moeten deze deelproblemen in de genoemde volgorde worden aangepakt. Daarom concentreren we ons in dit doctoraatsonderzoek op het eerste en belangrijkste probleem in de rij:

het voorspellen van draadlengtedistributies en afgeleide eigenschappen voor individuele schakelingen en voor specifieke keuzes m.b.t. de architectuur, en dit voor schakelingen waarin enkel verbindingen voorkomen die slechts twee bouwblokken verbinden.

Om dit doel te bereiken doen we eerst een grondige analyse van de twee basistechnieken waarop het meeste bestaand werk in dit domein gebaseerd is: de schattingstechniek van Donath en die van Davis (hoofdstuk 3). Beide worden geëvalueerd op basis van de mate waarin ze kunnen leiden tot betrouwbare *a priori* voorspellingen voor individuele schakelingen en op basis van hun flexibiliteit met betrekking tot de architectuur.

Zowel de techniek van Donath als die van Davis steunen op de *regel van Rent* als model voor de interconnectietopologie van de schakeling. Uit onze analyse blijkt dat de nauwkeurigheid en de interpretatie van deze empirische regel essentieel is voor de kwaliteit en zelfs de

bruikbaarheid van beide modellen om onze doelstellingen te bereiken. Daarom is hoofdstuk 4 volledig gewijd aan deze regel. We gaan op zoek naar nauwkeurigere modellen om ze te vervangen en trachten op die manier een aantal beperkingen van de technieken van Donath en Davis op te vangen.

Uit dit tweede analytische hoofdstuk moeten we echter concluderen dat de interpretatie van de regel van Rent die ten grondslag ligt aan Davis' techniek deze totaal onbruikbaar maakt voor het doel dat we in dit werk willen bereiken. Hoewel er aan Donaths techniek, in zijn oorspronkelijke vorm, een aantal beperkingen zijn, lijkt ze op dit ogenblik toch de meest geschikte basis om tot de gewenste voorspellingen te komen.

In hoofdstuk 5 stellen we verschillende aanpassingen van Donaths model voor. Deze leiden uiteindelijk tot bijzonder betrouwbare schattingen, die heel hoge correlaties vertonen met experimentele resultaten. Ons uitgebreid model slaagt erin zowel de verschillen tussen individuele schakelingen als tussen verschillende architecturen nauwkeurig te voorspellen.

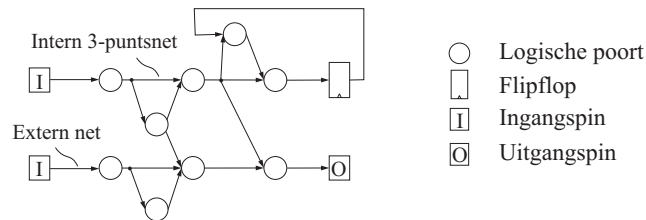
Om te komen tot voorspellingen van kostmetriekeken die sterk afhankelijk zijn van de interconnectielengtes, vormen de voorspelde draadlengtedistributies slechts een eerste stap. In hoofdstuk 6 trachten we de minimale klokperiode te voorspellen. Deze is in steeds grotere mate afhankelijk van de interconnectievertragingen, waardoor ook hier de bestaande modellen te kort schieten. Een bijkomende moeilijkheid is dat de minimale klokperiode afhankelijk is van de grootste combinatorische vertraging die in het uiteindelijke ontwerp optreedt. Door dit expliciet, op een probabilistische manier in rekening te brengen, slagen we erin zowel de betrouwbaarheid als de nauwkeurigheid van de bestaande schattingen te verbeteren.

In het laatste hoofdstuk van dit werk bespreken we de volgende problemen die volgens ons binnen dit onderzoeksthema moeten aangepakt worden, en hoe ons werk de oplossing van deze problemen dichterbij heeft gebracht.

De belangrijkste resultaten uit dit doctoraatsonderzoek werden voorgesteld op internationale conferenties en gedeeltelijk gepubliceerd in of opgestuurd naar internationale tijdschriften. Het theoretisch werk uit hoofdstuk 4 werd samengevat in [DSVC03a] en in [DVSVC01, DVSVC03]. De analyse van en uitbreidingen aan Donaths schattingsmodel (delen uit hoofdstukken 3 en 5) zijn te vinden

in [DVSVC02, DSVC03b] en het model voor de voorspelling van de minimale klokperiode (hoofdstuk 6) werd voorgesteld in [DSVC02].

Voor ons werk rond de predictie van draadlengtedistributies konden we voortbouwen op de ervaring rond dit thema die reeds in de onderzoeksgroep aanwezig was (hoofdzakelijk het werk van Herwig Van Marck [VM00] en Dirk Stroobandt [Str01]). De discussies en de uitwisseling van ideeën met Peter Verplaetse waren eveneens bijzonder verrijkend. Hij concentreerde zich in zijn doctoraal onderzoek hoofdzakelijk op het bestuderen en modelleren van de topologische eigenschappen van digitale schakelingen [Ver03]. Voor het onderzoek met betrekking tot het voorspellen van de minimale klokperiode hebben we kunnen putten uit de ervaring die werd opgedaan in eerder onderzoekswerk rond optische inter-chipverbindingen voor FPGAs [DVVC99a, DVVC99b, VCVMD99, DVVC00, DBMVC01].



Figuur 2.4: grafische voorstelling van een circuitgraaf, met aanduiding van de verschillende elementen.

2 Circuitgrafen en hun plaatsing

In hoofdstuk 2 worden de modellen voor de schakeling, de architectuur en het fysisch ontwerp besproken die ten grondslag liggen aan dit werk.

2.1 Circuitgrafen

Netten, blokken en pinnen

Zoals reeds besproken, willen we interconnectielengtes voorspellen op basis van, onder andere, de belangrijkste eigenschappen van een netlijst (figuur 2.4). Deze bestaat uit een set bouwblokken enerzijds en de manier waarop ze met elkaar verbonden zijn (de *interconnectietopologie*) anderzijds. De wiskundige voorstelling van een dergelijke netlijst is een graaf (wanneer enkel verbindingen tussen twee blokken zijn toegelaten) of een hypergraaf (als ook verbindingen tussen meer dan twee blokken kunnen bestaan).

Voor verbindingen wordt ook de term *netten* gebruikt. Netten die slechts twee blokken verbinden worden tweepuntsnetten genoemd, terwijl de andere meerpuntsnetten zijn. In dit werk is bijna uitsluitend sprake van tweepuntsnetten. Deze kunnen *intern* zijn, d.w.z. dat ze twee blokken van de schakeling verbinden, maar er bestaan ook *externe* netten, die de verbinding vormen tussen één blok en een ingangs- of uitgangssignaal van de schakeling.

Pinnen zijn de verbindingspunten tussen een net en een blok. Het gemiddeld aantal netten waarmee een blok verbonden is, is één van de karakteristieke parameters van een netlijst.

Voorstelling van het tijdsgedrag

Om ook het tijdsgedrag van synchrone schakelingen te modelleren (hoofdstuk 6), moeten de verbindingen of *netten* gericht zijn en moet een onderscheid gemaakt worden tussen combinatorische blokken (bijvoorbeeld logische poorten) en sequentiële blokken (bijvoorbeeld flipflops).

Modules

Een module van een schakeling is een subset van de blokken. De modulegrootte is het aantal blokken dat de module omvat. Het aantal netten dat blokken binnen de module verbindt met blokken erbuiten of met een ingangs- of een uitgangspin wordt het aantal pinnen van de module genoemd.

Een module kan ook bestaan uit één enkel blok of uit de gehele schakeling. Het aantal pinnen van de gehele schakeling wordt aangegeven door T_{ext} . Het is gelijk aan het totaal aantal ingangs- en uitgangssignalen van de schakeling.

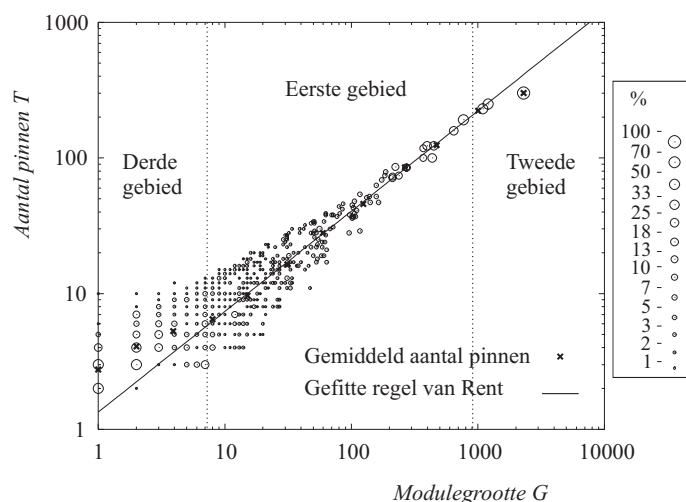
2.2 De regel van Rent

Een van de oudste en meest gebruikte technieken om snel tot een redelijk goede plaatsing van een schakeling te komen is het gebruik van recursieve partitionering. De schakeling wordt in twee of vier min of meer gelijke delen gesplitst op zo'n manier dat zo weinig mogelijk netten worden “doorgesneden”. Hetzelfde gebeurt met de architectuur en de modules van de schakeling worden aan de delen van de architectuur toegewezen. Dit proces wordt recursief herhaald, zodat men steeds kleinere modules krijgt.

In de jaren 1960 ontdekte E.F. Rent, een werknemer van IBM, dat het verband tussen het aantal pinnen T van die verschillende modules en hun grootte G kan benaderd worden door een machtwet:

$$T = tG^p. \quad (2.1)$$

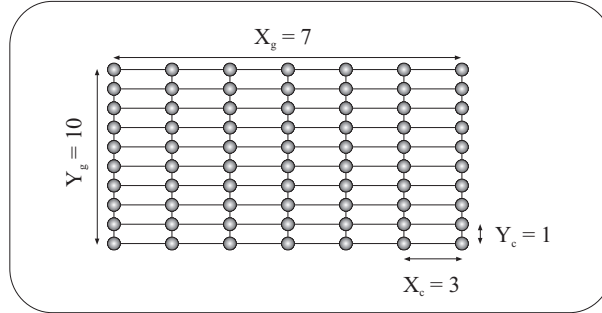
Figuur 2.5 toont een typisch voorbeeld van dit verband en de benadering door een machtwet. De parameters t en p worden de *Rentcoëfficiënt* en de *Rentexponent* genoemd. De regel van Rent is nog steeds een veel gebruikt model voor de complexiteit van de interconnectietopologie van digitale schakelingen.



Figuur 2.5: Voorbeeld van het verband tussen het aantal pinnen en de modulegrootte dat men vindt door een schakeling recursief te partitioneren en de benadering van dit verband door een machtwet.

Als het de bedoeling is om de eigenschappen van specifieke grafen zo goed mogelijk te benaderen, moeten t en p worden geschat door een fitprocedure. Hoewel deze benadering redelijk goed is voor het grootste gedeelte van de modules (het *eerste gebied* van de regel van Rent), treden er, zoals in figuur 2.5 te zien is, meestal afwijkingen op voor hele grote modules en voor hele kleine modules. Deze afwijkingen worden ook wel het *tweede gebied* en het *derde gebied* van de regel van Rent genoemd. Het opgemeten verband tussen T en G , dat door de regel van Rent wordt benaderd, wordt de *Rentkarakteristiek* genoemd.

Hoewel de regel van Rent oorspronkelijk werd gedefinieerd aan de hand van het recursief partitioneren van een circuitgraaf, zijn er nog veel andere manieren om modules te genereren. Het is bijvoorbeeld mogelijk om een gebied af te bakenen in een geplaatste schakeling en een module te vormen met alle blokken die zich binnen dat gebied bevinden. Zoals wordt besproken in hoofdstukken 3 en 4 heeft men de regel van Rent ook gebruikt om het aantal pinnen van dit soort modules te modelleren. We noemen deze benadering de *geometrische interpretatie* van de regel van Rent.



Figuur 2.6: Schematische weergave van het architectuurmodel dat in dit werk gebruikt wordt.

2.3 De architectuur

In de meeste technieken voor het schatten van interconnectielengtes wordt de architectuur gemodelleerd als een regelmatig rooster. Daarbij wordt verondersteld dat de blokken van de schakeling enkel op de roosterpunten geplaatst worden. Figuur 2.6 toont een voorbeeld van zo'n architectuurmodel. In dit werk worden enkel tweedimensionale architecturen beschouwd. Daarin kan het aantal roosterpunten in de horizontale richting X_g verschillen van het aantal roosterpunten in de verticale richting Y_g . Fysische afmetingen van de blokken, X_c en Y_c kunnen ook verschillen. Dit wordt in het rooster voorgesteld door in elke richting een verschillende eenheidsafstand te gebruiken.

In de meeste chips heeft elke routeringslaag één van twee orthogonale richtingen als voorkeursrichting. In de architectuur wordt daarom de afstand tussen twee roosterpunten met coördinaten (x_i, y_i) en (x_j, y_j) gemeten met een Manhattan-metriek:

$$d_{i,j} = |x_i - x_j| + |y_i - y_j|. \quad (2.2)$$

Interconnectielengte wordt eveneens uitgedrukt in Manhattan-eenheden.

2.4 Het plaatsingsmodel

Bij het voorspellen van interconnectielengtes wordt plaatsing in de zonet beschreven roosterarchitectuur beschouwd als model voor het volledig fysisch ontwerp. Bovendien wordt verondersteld dat tijdens deze

plaatsing de totale interconnectielengte (van de tweepuntsnetten) geminimaliseerd wordt. Het plaatsingsproces kan dus als volgt gedefinieerd worden:

plaats elk blok uit de netlijst op een unieke roosterpositie in het plaatsingsrooster, zó dat de totale verbindingslengte C zo klein mogelijk is.

Een dergelijk plaatsingsproces is *homogeen* met betrekking tot de interconnecties, in de zin dat iedere verbinding een even grote rol speelt in de totale kost C . Aan elke verbinding wordt dus even veel moeite besteed.

Het resultaat van een plaatsing, met betrekking tot interconnecties, kan op verschillende niveaus van detail worden beschouwd. Van globaal naar detaillistisch krijgen we:

- (i) de totale interconnectielengte,
- (ii) het aantal verbindingen van elke lengte,
- (iii) de lengtes van elke individuele verbinding en
- (iv) de gebieden waardoor deze verbindingen gerouteerd moeten worden.

In dit werk beogen we enkel voorspellingen van de eerste twee punten in deze lijst. Kostmetriecken die veeleer bepaald worden door de eigenschappen van een zeer beperkt aantal zeer specifieke verbindingen zullen uit deze schattingen dus niet of nauwelijks afgeleid worden.

Wegens de complexiteit van de zoekruimte, wordt het plaatsingsprobleem meestal op een heuristische manier aangepakt. Bovendien worden veel beslissingen genomen op basis van toevalligheden (zoals de namen van de knopen en netten in de netlijst) of zelfs op basis van een in het programma ingebouwde randomgenerator. Daardoor is de oplossing van het plaatsingsprobleem voor een gegeven netlijsttopologie niet deterministisch. We zullen in dit werk dan ook een stochastische interpretatie geven aan dit resultaat. Wegens de homogene kostfunctie, zullen we ervan uitgaan dat alle verbindingen statistisch gelijk zijn. Dit impliceert dat

de lengtes van de individuele verbindingen na plaatsing gelijk verdeelde toevalsgrootheden zijn, wanneer men ze beschouwt over de volledige set van geoptimaliseerde plaatsingen.

Deze veronderstelling noemen we de *homogeniteitsveronderstelling*. Ze vormt expliciet de basis van ons werk, maar was ook impliciet aanwezig in het werk van alle andere auteurs die geen onderscheid maakten tussen individuele verbindingen. De distributie van de verbindingslengtes noemen we de *draadlengtedistributie*.

In werkelijke plaatsingsprogramma's worden vaak nog andere kostmetrieken dan de totale verbindingslengte geoptimaliseerd. Dit heeft een weerslag op het bereikte plaatsingsresultaat. Nochtans gaan we er in dit werk van uit dat het resultaat van het hierboven beschreven plaatsingsproces, en dus ook onze schatting ervan, een maat is voor de *optimaliseerbaarheid* van de plaatsing van een bepaalde netlijst in een bepaalde architectuur. We kunnen dus hopen dat onze schattingen ook een goede correlatie vertonen met de resultaten die worden verkregen met andere kostmetrieken die te maken hebben met de interconnecties.

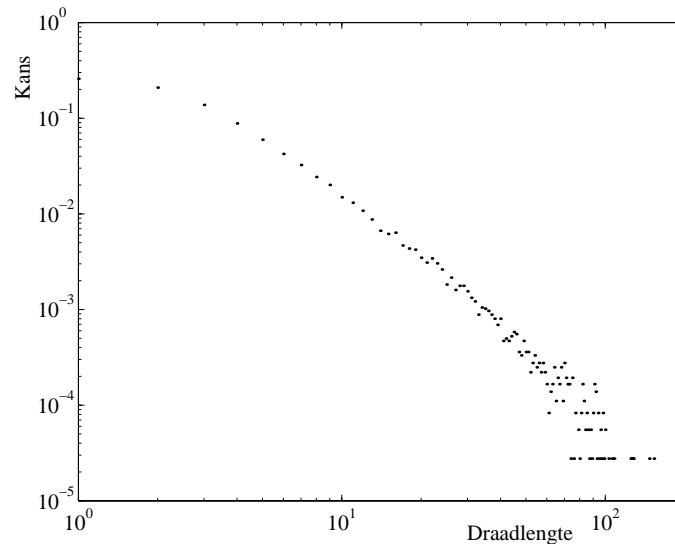
2.5 Een woordje over de experimenten

Om de in dit werk gemaakte schattingen te valideren, werd een groot aantal experimenten uitgevoerd. Hiervoor werden verschillende soorten testschakelingen (*benchmarks*) gebruikt.

Om in overeenstemming te zijn met de door ons gestelde randvoorwaarden, werden in deze schakelingen alle meerpuntsnetten vervangen door meerdere tweepuntsnetten. De plaatsingen gebeurden telkens in roosterarchitecturen. De draadlengtedistributie werd opgemeten door de lengteverdeling van de verbindingen, geëxtraheerd uit één of meerdere plaatsingen te normaliseren. Figuur 2.7 geeft een voorbeeld van zo'n opgemeten draadlengtedistributie.

Er kunnen bijzonder grote verschillen zijn tussen de resultaten van verschillende plaatsingsprogramma's onderling. Daarom worden in dit werk drie verschillende plaatsingsprogramma's gebruikt voor de experimenten. Onze resultaten tonen aan dat er, ondanks de grote verschillen tussen de benaderingen die in elke van deze programma's gebruikt worden, toch een sterke correlatie bestaat tussen de resultaten die deze programma's bereiken.

De schakelingen en plaatsingsprogramma's die voor de experimenten werden gebruikt staan in detail beschreven in appendix A van de Engelstalige tekst.



Figuur 2.7: Een typisch voorbeeld van een opgemeten draadlengtedistributie.

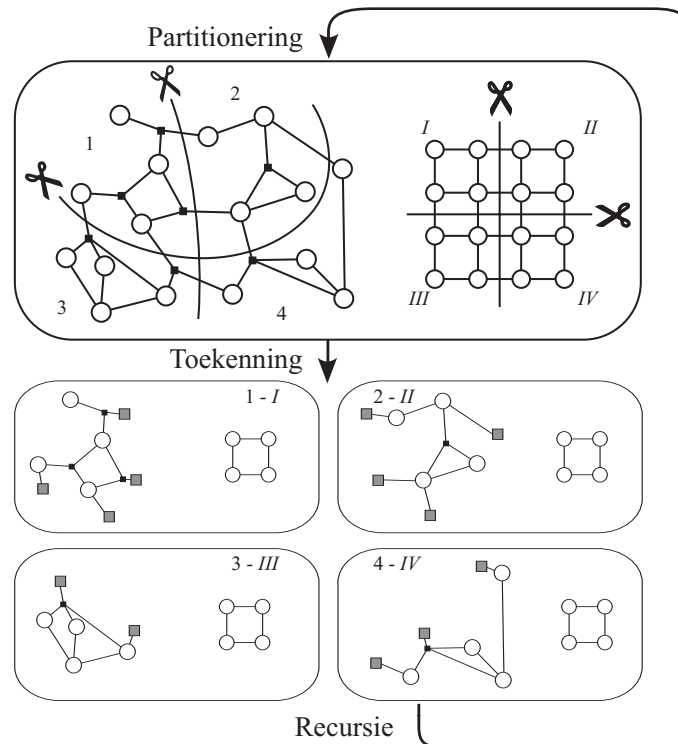
3 Basistechnieken voor de voorspelling van interconnectielengtes

In hoofdstuk 3 bestuderen we de twee basistechnieken voor het voorspellen van de draadlengtedistributie die ten grondslag liggen aan hedendaagse toepassingen van SLIP-technieken. Beide technieken, die fundamenteel verschillend zijn, worden geëvalueerd en hun belangrijkste beperkingen worden geïdentificeerd.

3.1 De schattingsmethode van Donath

In 1981 stelde Donath [Don81] voor om een plaatsing door recursieve partitionering wiskundig te modelleren. Daartoe beschouwde hij een vierkante roosterarchitectuur met gelijke eenheidsafstanden in beide richtingen. Het aantal roosterpunten in elke richting is een macht van twee. De schakeling bevat precies evenveel blokken als er roosterpunten zijn.

In zijn plaatsingsmodel (zie figuur 2.8) worden de schakeling en de architectuur in vier gelijke delen verdeeld. Het aantal verbindingen in de schakeling dat daarbij wordt “doorgesneden” wordt berekend op basis van de regel van Rent. Deze wordt dus expliciet verondersteld het



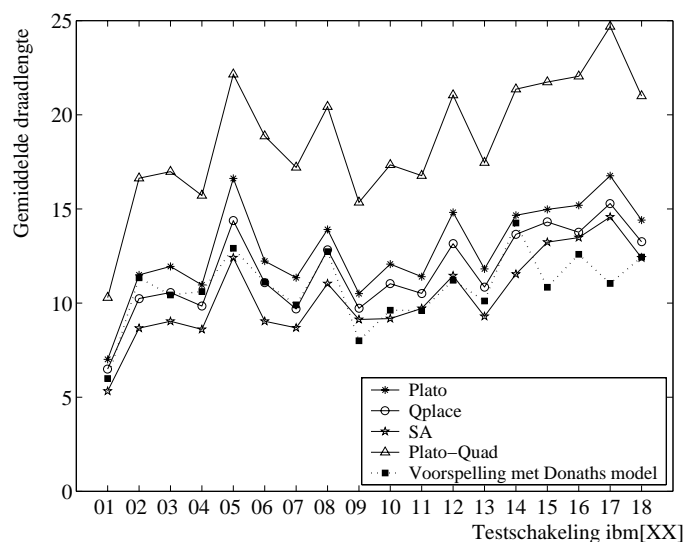
Figuur 2.8: Schematische weergave van de methode van Donath voor het schatten van draadlengtedistributies.

verband tussen pinnen en modulegrootte voor te stellen dat ontstaat bij het *recursief partitioneren* van de netlijst.

Om de lengtedistributie van de verbindingen te berekenen, veronderstelt Donath dat de modules van de schakeling op toevallige basis worden toegewezen aan de modules van de architectuur en dat de eindpunten van de gesneden verbindingen uniform verdeeld zijn over de architectuurmodules. De vermenigvuldiging van de resulterende distributie met het aantal gesneden netten geeft een schatting voor het aantal netten van elke lengte op dit hiërarchieniveau.

Deze bewerkingen worden recursief herhaald tot wanneer de modules niet verder verdeeld kunnen worden. De gevonden netlengteverdelingen van de verschillende hiërarchieniveaus worden tenslotte opgeteld om tot de globale lengteverdeling en (na normalisatie) draadlengtedistributie te komen.

We hebben eerst experimenteel vastgesteld dat Donaths methode



Figuur 2.9: Gemiddelde draadlengtes uit Donaths model, vergeleken met opgemeten gemiddelde draadlengtes. Het programma `Plato-quad` benadert Donaths plaatsingsmodel zo dicht mogelijk, terwijl de andere drie programma's (`Plato` en `SA`) beter benaderen wat in echte plaatsingsprogramma's gebeurt. `Qplace` is een commercieel plaatsingsprogramma.

wel degelijk resulteert in goede voorspellingen wanneer aan alle gestelde randvoorwaarden met betrekking tot de architectuur, de schakeling en de plaatsingsmethode voldaan is. In het bijzonder werd deze validatie uitgevoerd met synthetische testschakelingen waarvoor de regel van Rent perfect geldt.

Wanneer we Donaths methode echter toepassen op meer realistische schakelingen, waarvoor de regel van Rent slechts een ruwe benadering is, dan blijkt er slechts een matige correlatie te zijn tussen de voorspelde en gemeten gemiddelde draadlengten. Figuur 2.9 vergelijkt voor een aantal realistische testschakelingen de gemiddelde draadlengte die volgt uit Donaths model met de resultaten van verschillende plaatsingsprogramma's. Het eerste, `Plato-quad`, is een plaatsingsproces dat Donaths plaatsingsmodel zo goed mogelijk benadert. De andere drie leiden tot veel beter geoptimaliseerde plaatsingen. Vergeleken met het plaatsingsproces dat het nabootst, resulteert Donaths model dus bovendien in grote onderschattingen van de gemiddelde draadlengte.

Verschillende uitbreidingen aan Donaths model werden voorge-

steld door, onder andere, Van Marck [VM00] en Stroobandt [Str01]. We vermelden hier enkel de uitbreidingen die in de context van ons werk het belangrijkst zijn.

Van Marck paste de techniek toe op niet-vierkante tweedimensionale en driedimensionale plaatsingsroosters, met verschillende eenheidsafstanden in elke richting. Het aantal roosterpunten in elke richting was echter nog steeds een macht van twee en de schakeling werd verondersteld precies even groot te zijn als de architectuur.

In Donaths model wordt de plaatsing enkel geoptimaliseerd door het zo goed mogelijk partitioneren van de schakeling. Stroobandt stelde voor om, in de plaats van een uniforme verdeling van de eindpunten van de verbindingen, een *bezettingskans* te gebruiken die uitdrukt dat ook binnen een hiërarchieniveau de verbindingslengtes geminimaliseerd zijn.

Besluit met betrekking tot Donaths model

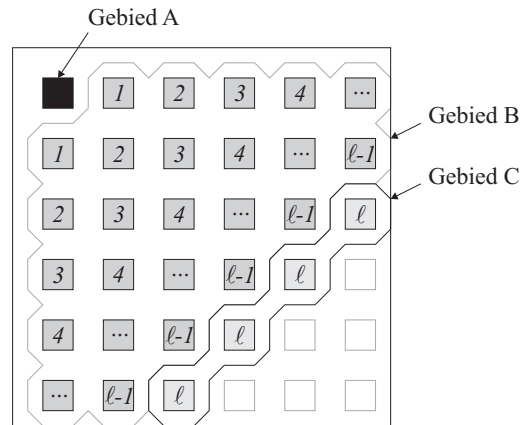
Het grootste verschil tussen de veronderstellingen in Donaths model en de experimenten met `Plato-quad` is de afwijking die de regel van Rent vertoont ten opzichte van de Rentkarakteristiek. Daarom kunnen we vermoeden dat deze afwijking de oorzaak is van de slechte correlatie tussen voorspellingen en experimentele waarden.

Bovendien laat Donaths model, zelfs met de uitbreidingen voorgesteld door Van Marck, slechts een beperkte variatie toe in het architectuurmodel.

3.2 De schattingsmethode van Davis

De schattingsmethode van Davis is gebaseerd op de geometrische interpretatie van de regel van Rent. Hij gebruikt deze regel immers om het aantal pinnen te berekenen voor modules die ontstaan door een gebied af te bakenen in een geplaatste schakeling.

Voor elk roosterpunt in de architectuur definieert Davis drie types gebieden in het plaatsingsrooster (zie figuur 2.10). Het gebied *A* is het betreffende roosterpunt zelf, gebied *B* omvat alle roosterpunten die minder dan een afstand ℓ van dit roosterpunt verwijderd zijn en het gebied *C* omvat de roosterpunten die zich precies op een afstand ℓ ervan bevinden. Het aantal verbindingen van lengte ℓ , verbonden met het centrale roosterpunt, is dan het aantal verbindingen tussen gebie-



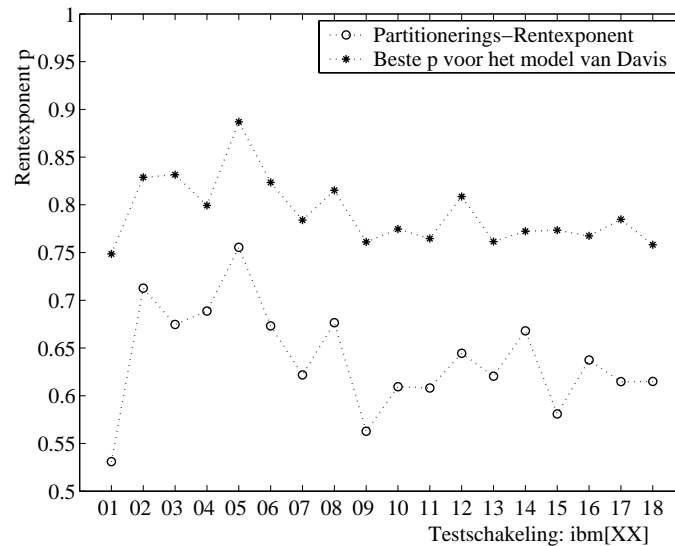
Figuur 2.10: De verschillende types gebieden die gebruikt worden in de methode van Davis.

den A en C .

De groottes van de gebieden A en B , evenals van de samengestelde gebieden AB , BC en ABC , kunnen berekend worden als functie van de afstand ℓ en van de positie van het centrale roosterpunt. Het aantal pinnen van de modules die erdoor worden afgebakend wordt berekend uit de regel van Rent. Uit deze aantallen kan het gezochte aantal verbindingen berekend worden. Door deze berekening vervolgens te herhalen voor elke lengte en elk roosterpunt en de gevonden aantallen op te tellen, wordt (weerom na normalisatie) de totale draadlengtedistributie verkregen.

In deze methode wordt voor alle modules eenzelfde waarde van de Rentexponent p gebruikt, onafhankelijk van de vorm of de positie van het betreffende gebied. Bovendien is het helemaal niet duidelijk hoe men deze waarde *a priori* kan bepalen. Is ze gelijk aan de partitionerings-Rentexponent uit Donaths methode? In de meeste toepassingen van Davis' techniek worden geen schattingen gemaakt voor individuele schakelingen, maar voor klassen van schakelingen. Voor p wordt dan een typische waarde gebruikt, waarvan men veronderstelt dat ze representatief is voor, bijvoorbeeld, de ontwerpen uit een bepaald toepassingsgebied.

Hoewel Davis' schattingsmethode oorspronkelijk beschreven werd voor vierkante architecturen met gelijke eenheidsafstanden in elke richting, ligt de uitbreiding ervan naar rechthoekige roosters en verschillende eenheidsafstanden voor de hand. We hebben getest of deze me-



Figuur 2.11: Vergelijking tussen de Rentexponent zoals die gedefinieerd is in Donaths model en de gefitte waarde van p die in Davis' model het beste resultaat geeft.

thode kan resulteren in schattingen die goed correleren met plaatsingsresultaten voor verschillende architecturen. Aangezien we niet weten hoe de waarde van p moet bepaald worden, hebben we ze beschouwd als een parameter. Om betrouwbaar te zijn, moet er immers voor elke schakeling minstens één waarde van p bestaan waarvoor een goede correlatie bekomen wordt. Dit bleek inderdaad het geval te zijn. Zoals uit figuur 2.11 blijkt, verschillen de op die manier gefitte waarden van p sterk van de Rentexponenten die in Donaths' model gebruikt worden.

Besluit met betrekking tot Davis' model

Net als in Donaths model blijken ook in het model van Davis' de regel van Rent en de waarde van de Rentexponent essentieel te zijn. Het probleem is nu echter niet zo zeer de benadering die dit inhoudt, maar wel het feit dat we eenvoudigweg niet weten hoe p a priori kan worden bepaald. In deze methode wordt de regel van Rent immers gedefinieerd als het gevolg van een goede plaatsing. Bovendien kunnen we ons afvragen of het verband tussen het aantal pinnen en de modulegrootte bij geometrisch bepaalde modules wel onafhankelijk kan zijn van de vorm en de plaats van de module in het plaatsingsrooster.

Met betrekking tot de plaatsingsarchitectuur zijn er aan Davis' model geen beperkingen verbonden.

4 Het aantal pinnen van verschillende moduletypes

In hoofdstuk 4 gaan we op zoek naar antwoorden op de vragen omtrent de regel van Rent en de Rentkarakteristiek die aan het licht kwamen in hoofdstuk 3.

Eerst en vooral tonen we aan, aan de hand van metingen, dat men modules op heel veel manieren kan genereren, en dat elke generatiemethode resulteert in een andere Rentkarakteristiek. Vervolgens bestuderen we enerzijds Rentkarakteristieken die ontstaan door recursief partitioneren (*partitionerings-Rentkarakteristieken*) en anderzijds Rentkarakteristieken op basis van modules die ontstaan door een gebied af te bakenen in een geplaatste schakeling (*geometrische Rentkarakteristieken*). Eenvoudige metingen tonen reeds aan dat, voor een zelfde schakeling, beide types Rentkarakteristieken sterk kunnen verschillen.

4.1 Partitionerings-Rentkarakteristieken

De Rentkarakteristiek die nodig is voor Donaths schattingstechniek kan zonder problemen a priori opgemeten worden. Nochtans gaat dit ten koste van veel rekenkracht. Er moet immers een sequentie van partitioneringen worden uitgevoerd, en dit zijn eveneens harde optimalisatieproblemen. Men kan deze rekentijd beperken door slechts een beperkt aantal partitioneringen uit te voeren. De resterende punten van de Rentkarakteristiek kunnen dan door interpolatie worden berekend. Dit geeft echte vrij grote afwijkingen tegenover een volledig opgemeten Rentkarakteristiek.

Op basis van een model voor de partitionerings-Rentkarakteristiek met één of enkele parameters zou allicht een meer nauwkeurige schatting mogelijk zijn. Een dergelijk model kan dan meteen ook de regel van Rent, die erg onnauwkeurig is, vervangen.

In zijn doctoraat stelde Verplaetse [Ver03] een model voor dat het aantal gesneden netten modelleert bij recursief bipartitioneren. Zijn model is gebaseerd op een stelsel recursievergelijkingen en is niet analytisch oplosbaar. Een numerieke oplossing is met de tegenwoordig

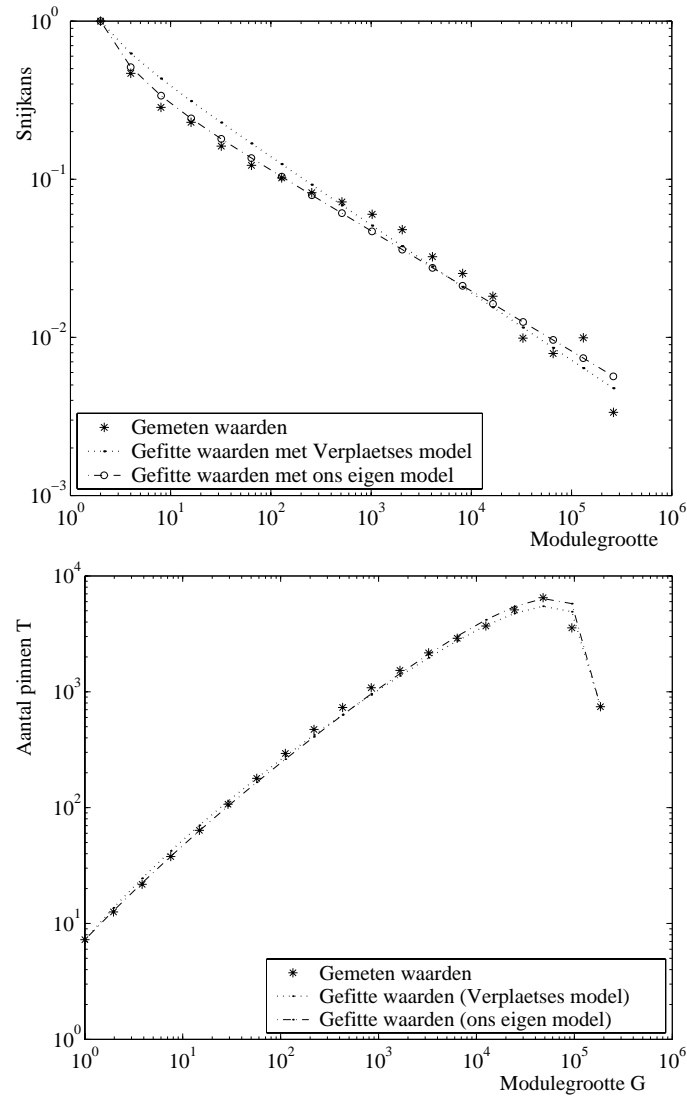
beschikbare rekenkracht echter geen probleem meer.

Bij het in twee delen van een module, kunnen het aantal pinnen, het aantal interne netten en de grootte van de deelmodules uitgedrukt worden als functie van dezelfde grootheden voor de oorspronkelijke module. Daarbij wordt de optimalisatie uitgedrukt door de fractie van de interne netten die wordt gesneden (de *snijkans*). Verplaetse stelt dat deze snijkans afhankelijk is van de modulegrootte en stelt een uitdrukking voor met twee parameters, τ en ϵ . Wanneer deze snijkans wordt ingevuld in de recursiebetrekkingen en de parameters worden bepaald door middel van een fitprocedure, benadert het resultaat de opgemeten partitionerings-Rentkarakteristieken van realistische schakelingen vrij goed. Deze overeenkomst neemt echter snel af wanneer slechts een beperkt aantal partitioneringsniveaus worden gebruikt voor het fitten.

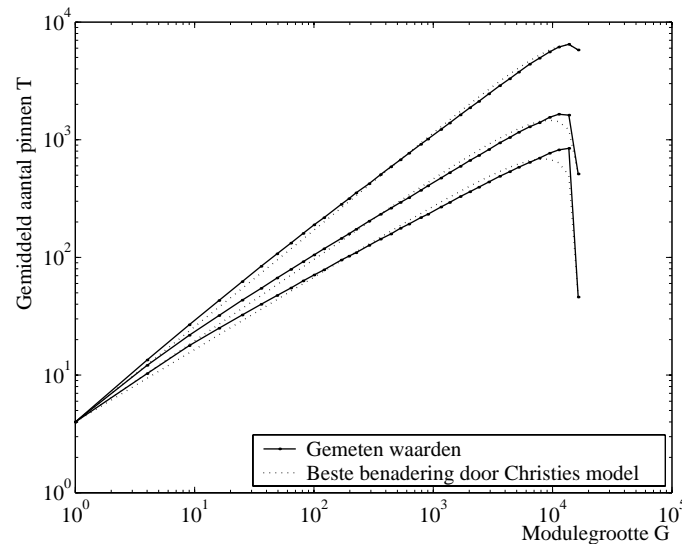
We hebben, onafhankelijk van Verplaetse, een alternatieve uitdrukking voor de snijkans voorgesteld (het *beta*-model). Deze bevat slechts één parameter, β , en modelleert de snijkans en de Rentkarakteristiek minstens even goed als Verplaetses model. Bovendien blijkt het ook te leiden tot zeer goede schattingen als slechts drie of vier partitioneringsniveaus gebruikt worden. Figuur 2.12 geeft een voorbeeld van deze snijkansen en de partitionerings-Rentkarakteristiek, opgemeten voor een realistische schakeling. Het best passende resultaat van Verplaetses model en van onze eigen alternatieve uitdrukking voor de snijkans zijn eveneens weergegeven.

Verplaetses idee om de snijkans te modelleren, in plaats van rechtstreeks het aantal pinnen per module, is volgens ons de juiste manier om te komen tot meer onderbouwde theoretische modellen voor de partitionerings-Rentkarakteristiek. Deze kunnen dan bijvoorbeeld gebruikt worden om interconnectieschattingen te maken op een moment tijdens het ontwerpproces waarop de volledige netlijst van het ontwerp nog niet beschikbaar is. Daarvoor moet dan wel worden onderzocht wat typische waarden van de gebruikte parameters zijn, en dit in functie van het toepassingsdomein.

Beide tot nu toe voorgestelde modellen voor de snijkans zijn eerder empirisch. In de toekomst zou zeker verder moeten gezocht worden naar een meer theoretisch gefundeerde uitdrukking voor deze kans.



Figuur 2.12: (boven) Opgemeten snijkansen voor een realistische schakeling en berekende waarden met gefitte parameters τ en ϵ (voor Verplaetses model) en β (voor ons eigen model; (onder) gemeten partitionerings-Rentkarakteristiek en Rentkarakteristieken berekend door Verplaetses recursievergelijkingen, éénmaal met Verplaetses uitdrukking voor de snijkans en éénmaal met ons eigen alternatief.



Figuur 2.13: Opgemeten geometrische Rentkarakteristieken voor enkele synthetische testschakelingen en hun benadering door Christies model.

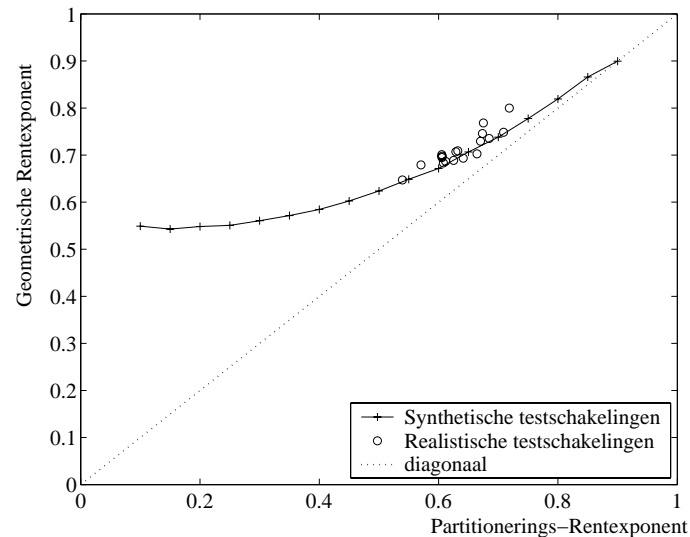
4.2 Geometrische Rentkarakteristieken

Het model van Christie

Ook voor geometrische Rentkarakteristieken gaan we op zoek naar een beter model dan de regel van Rent. In eerste instantie bespreken we een model dat werd voorgesteld door Christie [Chr01]. Hij beschouwt een geometrische module en beschrijft wat er gebeurt met het aantal pinnen van die module als ze een heel klein beetje groter gemaakt wordt. Op die manier komt hij tot een differentiaalvergelijking die het verband tussen het aantal pinnen en de modulegrootte uitdrukt. De oplossing van deze vergelijking geeft een uitdrukking met één enkele parameter ρ , die een gelijkaardige rol speelt als de snijkans in het model voor partitionerings-Rentkarakteristieken.

Net als vroeger de Rentexponent, zorgt de parameter ρ voor het exponentiële schalingsgedrag van de Rentkarakteristiek. Hij kan opnieuw zó gekozen worden dat het resultaat een vrij goede benadering vormt van opgemeten geometrische Rentkarakteristieken. Figuur 2.13 geeft een voorbeeld van geometrische Rentkarakteristieken voor synthetische schakelingen en hun benadering door Christies model.

Uit de figuur blijken er toch systematisch enkele afwijkingen op te



Figuur 2.14: Geometrische Rentexponent als functie van de partitionerings-Rentexponent: opgemeten verband voor enkele synthetische en realistische testschakelingen.

treten tussen de metingen en het model. Vermoedelijk zijn deze ondermeer te wijten aan het feit dat in Christies model, de parameter ρ onafhankelijk is van de modulegrootte. Dit was in ieder geval niet zo voor de hoger besproken snijkansen. Om deze hypothese te testen, stellen we zelf een alternatief voor. Dit geeft inderdaad aan waar de afwijkingen in Christies model vandaan komen.

Een nauwkeurig enumeratiemodel

Hoewel we er op basis van Christies model in slagen om geometrische Rentkarakteristieken beter te benaderen, hebben we nog steeds geen enkele aanwijzing omtrent het a priori bepalen van de nodige parameters. Bovendien weten we ook nog niet of één enkele Rentkarakteristiek wel volstaat voor het beschrijven van de verschillende modules die in Davis' techniek gebruikt worden.

Daarom stellen we een beschrijvend enumeratiemodel op om het aantal pinnen van een module te berekenen aan de hand van de draadlengtedistributie. Op basis van dit model slagen we erin enkele fundamentele eigenschappen van geometrische Rentkarakteristieken aan te tonen. Samengevat stellen we het volgende vast:

- het schalingsgedrag van geometrische Rentkarakteristieken verschilt van dat van partitionerings-Rentkarakteristieken. Waar de eerste benaderd kunnen worden door een machtwet met Rent-exponent $p \in [0, 1]$ is de exponent voor geometrische karakteristieken beperkt tot het interval $[0.5, 1]$; figuur 2.14 geeft het opgemeten verband weer tussen beide exponenten voor verschillende testschakelingen;
- de ligging van een geometrische Rentkarakteristiek is afhankelijk van de vorm van de beschouwde gebieden; het schalingsgedrag (de exponent) is hier echter relatief ongevoelig aan;
- de geometrische Rentkarakteristieken worden bepaald door de draadlengtedistributie; het is dus een omgekeerde redenering om ze te gebruiken om deze nauwkeurig te voorspellen.

Tot slot stellen we experimenteel vast dat, zelfs indien we nauwkeurige, *a priori* voorspellingen konden doen van de geometrische Rentkarakteristieken, Davis' techniek nog steeds onvoldoende nauwkeurig zou zijn. De toepassing van één enkele Rentkarakteristiek op modules met sterk verschillende vormen geeft immers te grote fouten in het model.

4.3 Algemeen besluit uit deze analyse

We kunnen besluiten dat Davis' model geen goede basis is om *a priori* nauwkeurige en betrouwbare voorspellingen te doen van de interconnectie-eigenschappen van individuele schakelingen.

Donaths model daarentegen laat dit misschien wel toe, op voorwaarde dat de regel van Rent vervangen wordt door een Rentkarakteristiek en dat de beperkingen met betrekking tot de architectuur worden opgeheven.

5 Een meer nauwkeurige voorspelling van interconnectielengtes

5.1 Verbetering van de betrouwbaarheid

In hoofdstuk 5 heffen we stap voor stap de belangrijkste beperkingen van Donaths model op. In eerste instantie vervangen we de regel

van Rent door de opgemeten partitionerings-Rentkarakteristiek. Deze wordt bovendien getransformeerd (“verschoven”) om te compenseren voor het feit dat in echte plaatsingsexperimenten de schakeling meestal niet precies even groot is als de architectuur.

Vervolgens pakken we de beperkingen van Donaths architectuurmodel aan. De ergste beperking, het feit dat het aantal roosterposities in elke richting een macht van twee is, is te wijten aan de eis dat de architectuurmodules steeds in perfect gelijke delen moeten gesplitst worden. In onze aangepaste versie van het model laten we toe dat de afmetingen van de delen lichtjes verschillen. Daardoor kunnen we roosters behandelen met om het even welke afmetingen en we laten ook een ruime vrijheid toe in de verschillende eenheidsafstanden.

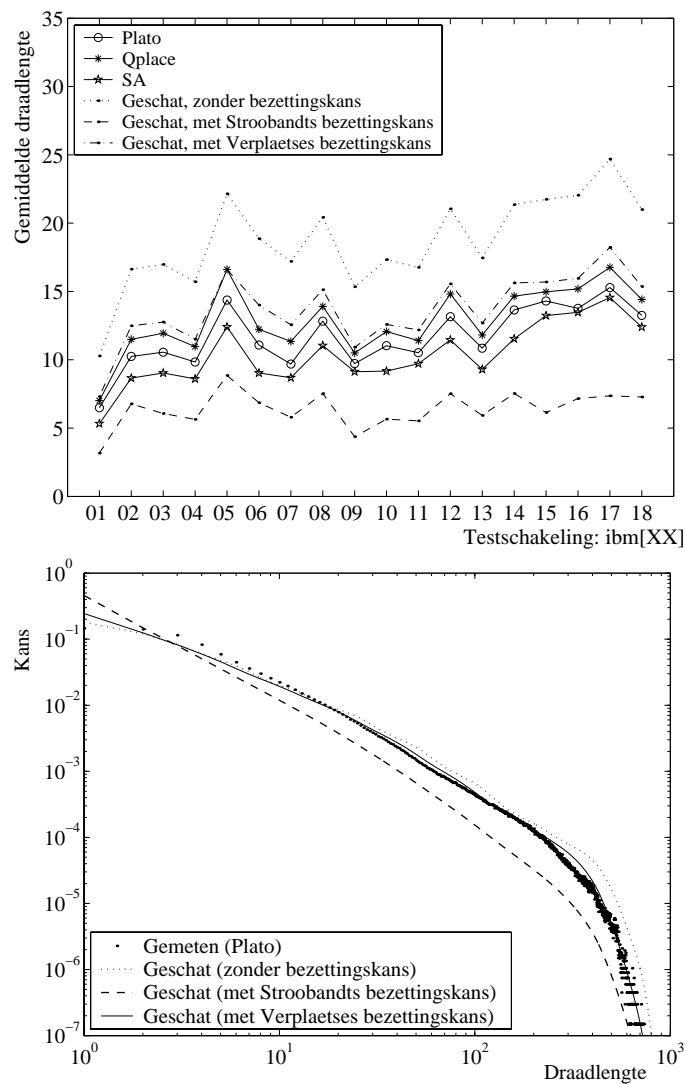
Met deze uitbreidingen komen we tot schattingen voor de draadlengtedistributie waarvan de gemiddelde waarden, over verschillende testschakelingen heen, bijzonder goed correleren met de experimenteel gemeten waarden. Deze correlaties liggen nu tussen 0.952 en 0.985, afhankelijk van het plaatsingsprogramma waarmee de experimentele waarden gegenereerd worden. De correlaties tussen de voorspelde gemiddelde draadlengtes uit Donaths oorspronkelijk model en dezelfde plaatsingsresultaten lagen tussen 0.697 en 0.804. Deze sterk verbeterde correlatie is hoofdzakelijk te wijten aan het gebruik van de Rentkarakteristiek in plaats van de regel van Rent. We zijn met deze uitbreidingen dus in staat om het onderscheid tussen verschillende interconnectietopologieën met grote betrouwbaarheid te voorspellen.

Ook de invloed van variaties in de architectuur wordt door ons uitgebreid model goed voorspeld. Voor een set van 70 verschillende architecturen, vinden we correlaties van 0.964 tot 0.999, afhankelijk van de testschakeling.

5.2 Verbetering van de nauwkeurigheid

Hoewel we met ons model nu een heel hoge betrouwbaarheid bereiken, vormen de geschatte gemiddelde draadlengtes nog een heel grote overschatting, in vergelijking met de gemeten waarden. Dit komt door de beperkte mate van optimalisatie die in Donaths model aanwezig is (zie figuur 2.15). Om dit gedeeltelijk te verhelpen, maken we gebruik van het idee van Stroobandt [Str01] om een *bezettingskans* in het model op te nemen.

De uitdrukking voor deze bezettingskans die door Stroobandt zelf



Figuur 2.15: Vergelijking tussen schatting en experiment: (boven) gemiddelde draadlengtes voor verschillende testschakelingen en verschillende plaatsingsprogramma's; (onder) draadlengtedistributie uit een plaatsing met Plato voor één testschakeling. In beide figuren zijn de schattingen weergegeven zonder bezettingskans, evenals de schattingen die gebruik maken van de uitdrukkingen voor de bezettingskans van Stroobandt en Verplaetse.

werd voorgesteld leidt echter tot zeer sterke onderschattingen van de gemiddelde draadlengte (ook weergegeven op figuur 2.15). Bovendien gaat ook een gedeelte van de bereikte betrouwbaarheid (correlatie) erdoor verloren. Als alternatief beschouwen we een uitdrukking voor de bezettingskans die door Verplaetse werd voorgesteld [Ver03]. Deze modelleert de bijkomende optimalisatie die kan bereikt worden door binnen elk hiërarchieniveau de modules zó toe te kennen dat de totale draadlengte zo klein mogelijk is. In Donaths model werd verondersteld dat deze toekenning op toevallige basis gebeurde.

Na kleine aanpassingen om Verplaetses bezettingskans in ons uitgebreid model bruikbaar te maken, bekomen we schattingen die veel dichterbij de experimenten liggen, en dit nauwelijks ten koste van de betrouwbaarheid. In vergelijking met de resultaten van `Plato`, het plaatsingsprogramma dat het dichtst aanleunt bij Verplaetses plaatsingsmodel, wijkt de geschatte gemiddelde draadlengte nu voor de meeste schakelingen minder dan 10% af van de gemeten waarde. De bereikte schattingen worden vergeleken met de experimentele waarden in de bovenste grafiek van figuur 2.15. Als we bovendien de geschatte draadlengtedistributies vergelijken met die van `Plato`, dan blijken deze erg nauwkeurig te zijn voor verbindingen langer dan $\ell = 10$. Voor kortere verbindingen treden wel nog enkele systematische afwijkingen op (zie onderste grafiek van figuur 2.15 voor een typisch voorbeeld).

5.3 Gevoeligheid aan de gebruikte Rentkarakteristiek

Tot slot gaan we na in hoeverre de betrouwbaarheid en kwaliteit van onze schattingen gevoelig zijn aan de gebruikte Rentkarakteristiek. We stellen vast dat de variaties die te wijten zijn aan toevallige keuzes in het partitioneringsprogramma vrijwel geen invloed hebben.

Wanneer een Rentkarakteristiek gebruikt wordt die geschat werd op basis van een model voor de snijkans (hoofdstuk 4), treedt er wel een significant verlies aan betrouwbaarheid op. De gemaakte schattingen zijn echter nog steeds beduidend betrouwbaarder dan diegene die men vindt door de regel van Rent te gebruiken.

6 Voorspelling van de minimale klokperiode

6.1 De klokperiode en het traagste pad

Een combinatorisch pad in een schakeling is een gerichte opeenvolging van logische poorten en verbindingen, die start aan de uitgang van een geheugencel of flipflop en eindigt aan de ingang van zo'n cel. De vertraging van zo'n pad wordt bepaald door de som van de vertragingen van individuele poorten en verbindingen. Ze bepaalt hoe lang het duurt vooraleer het signaal op het einde van het pad (aan de ingang van de flipflop) een correcte eindwaarde heeft bereikt en stabiel is.

In een synchrone schakeling wordt de klokperiode bepaald door de grootste combinatorische padvertraging in de schakeling. Traditioneel wordt de klokperiode a priori geschat als de vertraging van één pad met maximale combinatorische paddiepte. Deze wordt dan berekend op basis van "gemiddelde" poorten en verbindingen van gemiddelde lengte.

Nochtans zijn er in de meeste schakelingen vrij veel paden met maximale diepte, het is enkel het maximum van de vertragingen over al deze paden dat de klokperiode bepaalt. Nu bovendien de vertragingen van de langste verbindingen soms de vertraging van één enkele poort overstijgen, kan het traagste pad eveneens minder dan de maximale diepte hebben. Om op een correcte manier de klokperiode te kunnen schatten dient dus rekening gehouden te worden met alle paden en met het feit dat de klokperiode een *maximum* is.

6.2 Een probabilistische benadering

In het kader van dit werk worden de vertragingen van de verbindingen, net zoals hun lengtes, beschouwd als gelijk verdeelde toevalsgrootheden. De vertragingsdistributie kan aan de hand van bestaande modellen berekend worden uit de draadlengtedistributie.

De vertraging van een pad is dan de som van een aantal vertragingen. De distributie daarvan kan eveneens berekend worden (door convolutie). Voor een aantal paden waarvan de vertragingen statistisch onafhankelijk zijn, d.w.z. paden die geen verbindingen gemeenschappelijk hebben, kan ook de distributie van het maximum van al hun vertragingen berekend worden.

Het probleem is echter dat de paden in digitale schakelingen totaal

niet onafhankelijk zijn. Om toch een schatting te maken van de globale maximale vertraging, stellen wij een *vertragingsequivalente graaftopologie* voor. Dit is een graaftopologie die enkel *onafhankelijke paden* bevat en evenveel verbindingen als de oorspronkelijke graaf, en waarvan de maximale vertraging die van de oorspronkelijke graaf benadert. Voor een dergelijke topologie kan immers wel de distributie van de maximale vertraging berekend worden. Voor het berekenen van deze topologie houden we rekening met de volledige padstructuur van de schakeling.

6.3 Resultaten en besluit

We hebben ons probabilistisch model opnieuw gevalideerd aan de hand van testschakelingen. De vertragingen werden berekend op basis van de technologische eigenschappen voor toekomstige technologieën die worden weergegeven in de ITRS (*International Roadmap on Semiconductors*) en vertragingmodellen uit de literatuur. In de opeenvolgende technologiegeneraties uit de ITRS wordt de invloed van interconnecties steeds belangrijker. We gaan dus ook na hoe de kwaliteit van onze en van de traditionele schatting evolueren doorheen deze generaties.

Wanneer we voor de schattingen de opgemeten draadlengtedistributie gebruiken, vinden we een correlatie tussen de resultaten van ons model en de metingen van 0.971, en dit voor verschillende technologiegeneraties. Voor de traditionele methode is deze correlatie initieel 0.954, maar ze daalt naarmate de interconnecties belangrijker worden en bereikt een laagste waarde van 0.948. Onze schatting is dus in ieder geval betrouwbaarder dan de traditionele.

Het grootste verschil ligt echter in de nauwkeurigheid. Waar de traditionele schatting de klokperiode gemiddeld 25% onderschatte, vinden we voor ons model een gemiddelde overschatting van rond de 9%.

Door middel van Monte-Carlosimulatie hebben we kunnen vaststellen dat de grootste over- en onderschattingen te wijten zijn aan de homogeniteitsveronderstelling. Dit is het geval voor beide methodes, aangezien beide op deze veronderstelling berusten. Voor een meer nauwkeurige en vooral een meer betrouwbare voorspelling van de klokperiode zou het dus wenselijk zijn om ook lokale afwijkingen van deze veronderstelling te kunnen voorspellen. Nochtans blijkt dit slechts echt noodzakelijk te zijn voor een beperkt aantal schakelingen.

Tot slot hebben we vastgesteld dat het voor deze toepassing nood-

zakelijk is om de draadlengtedistributie nauwkeurig te voorspellen. Het nemen van een maximum is immers heel gevoelig aan de vorm van de distributie. De kleine overschatting die gemaakt wordt in de voorspelde draadlengtedistributies vermindert de correlatie lichtjes (die daalt tot 0.966) en leidt tot een stijging van de gemiddelde fout tot ongeveer 14%.

Hoewel dit model zowel een conceptuele als een meetbare verbetering inhoudt in vergelijking met de traditionele benadering, is het duidelijk nog niet optimaal. Zowel verbeteringen aan de equivalente graaftopologie als de voorspelling van afwijkingen van de homogeniteitsveronderstelling moeten nog verder onderzocht worden.

7 Enkele ideeën voor verder onderzoek

In hoofdstuk 7 worden de belangrijkste thema's voor voortgezet onderzoek aangegeven. Voor veel van de gestelde onderzoeksvragen wordt ook een mogelijke strategie aangegeven om hun oplossing aan te vatten. In deze samenvatting beperken we ons tot het formuleren van de belangrijkste problemen.

7.1 Verfijningen van en uitbreidingen aan de voorgestelde modellen

Om te kunnen komen tot nog betere en betrouwbaardere schattingen van interconnectie-eigenschappen kunnen nog heel wat verbeteringen aangebracht worden aan de in dit werk voorgestelde modellen.

Rechtstreekse uitbreidingen

Vooreerst zijn er een aantal aspecten die kunnen aangepakt worden binnen het raamwerk van veronderstellingen waar we in dit werk van uitgegaan zijn:

- voorspelling van **de draadlengtedistributie**: er zou kunnen gezocht worden naar uitdrukkingen voor de bezettingskans die enkele parameters bevatten, waardoor ze kunnen aangepast worden om de prestaties van specifieke plaatsingsprogramma's te voorspellen; op die manier kan, naast een grote betrouwbaarheid, ook de nauwkeurigheid van de gemaakte schattingen verbeterd

worden;
ook wordt in de huidige uitdrukkingen voor de bezettingskans nog steeds de Rent-exponent gebruikt. Het zou beter zijn om ook hier de Rentkarakteristiek te kunnen gebruiken;

- voorspelling van **de klokperiode**: de voorgestelde manier om tot een equivalente graaftopologie te komen is redelijk eenvoudig en ad hoc; vermoedelijk kan een meer diepgaand onderzoek leiden tot een meer onderbouwde methode, die leidt tot een grotere betrouwbaarheid van de voorspellingen.

Uitbreidingen aan het onderliggend raamwerk

De belangrijkste en volgens ons meest dringende thema's voor voortgezet onderzoek vereisen wijzigingen aan het onderliggend raamwerk van veronderstellingen. De belangrijkste daarbij zijn het includeren van meerpuntsnetten in de modellen en het in rekening brengen van afwijkingen t.o.v. de homogeniteitsveronderstelling.

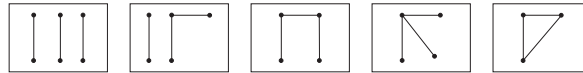
Het voorspellen van de lengtes van **meerpuntsnetten** is in het verleden altijd beschouwd als een moeilijk probleem; het feit dat ze er zijn en dat ze mee geoptimaliseerd worden in de kostfunctie heeft een belangrijke invloed op het plaatsingsresultaat. Hun lengtes kunnen dus niet zonder meer geschat worden uit de geschatte lengtedistributie van tweepuntsnetten. Bovendien is voor meerpuntsnetten de homogeniteitsveronderstelling niet langer vervuld.

Zoals is gebleken uit hoofdstuk 6, treden zelfs afwijkingen van de homogeniteitsveronderstelling op als enkel tweepuntsnetten beschouwd worden. Het bestuderen en modelleren van deze afwijkingen vergt zeker nog heel wat onderzoek.

7.2 Versteving van de theoretische onderbouw

In dit werk zijn we erin geslaagd om a priori meer betrouwbare en nauwkeurige schattingen te doen van interconnectie-eigenschappen. Deze zijn echter nog steeds afhankelijk van recursieve partitionering om de eigenschappen van de interconnectietopologie te extraheren. In zekere zin is dit een indirecte meting van deze eigenschappen, aangezien we verplicht zijn om één optimalisatieprobleem (partitionering) op te lossen om het resultaat van een ander (plaatsing) te schatten. Het

Interconnectiepatronen met drie netten:



Figuur 2.16: Mogelijke interconnectiepatronen met 3 verbindingen.

zou veel beter zijn om door rechtstreekse meting van topologische eigenschappen te komen tot hetzelfde resultaat.

Dit vereist echter een meer diepgaande kennis van de topologische eigenschappen van grafen en hypergrafen en van hun gedrag onder optimalisatie. Dit zijn onderwerpen die ook relevant zijn voor andere onderzoeksdomeinen. We zijn dan ook op zoek gegaan naar verbanden tussen het onderzoek dat in andere domeinen wordt verricht en onze eigen vraagstelling.

Eerst tonen we aan dat de zoekruimte van zowel het plaatsingsprobleem als het partitioneringsprobleem kunnen gerelateerd worden aan de interconnectietopologie van de schakeling. In het bijzonder is het belangrijk de frequentie te kennen waarmee bepaalde interconnectiepatronen in de graaf voorkomen. Figuur 2.16 toont enkele dergelijke interconnectiepatronen.

In de random-graaftheorie is men erin geslaagd om deze voorkomensfrequenties te modelleren voor random grafen, en dit enkel op basis van eenvoudig meetbare grootheden zoals het aantal blokken en het aantal verbindingen in de graaf. Helaas zijn circuitgrafen geen random grafen. Ze blijken daar zelfs van te verschillen op een manier die vergelijkbaar is met die van andere grafen uit uiteenlopende onderzoeksdomeinen die in de literatuur beschreven staan. Aan het trachten quantificeren en reproduceren van deze verschillen wordt veel aandacht besteed in een vrij recent multidisciplinair onderzoeksdomein rond complexe netwerken. De contacten tussen dit onderzoeksdomein en de wereld van digitaal ontwerp zijn echter nog vrijwel onbestaand.

Behalve de topologische eigenschappen van circuitgrafen, blijft ook hun gedrag bij optimalisatie een vraagteken. Het voorspellen van de snijkans bij optimale partitionering is tot nog toe enkel gelukt voor een bepaalde subset van random grafen. Deze oplossing werd gevonden door technieken te gebruiken uit de statistische mechanica. Nochtans zijn in de literatuur wel empirische resultaten te vinden voor sommige klassen van niet-random grafen. Het verloop van de snijkans als functie van de modulegrootte dat werd gevonden voor deze grafen, is zeer

gelijkaardig aan de modellen voor de snijkans die werden voorgesteld door Verplaetse en onszelf.

In het algemeen kunnen we stellen dat deze eerder theoretische problemen het meest efficiënt kunnen worden aangepakt in samenwerking met onderzoekers uit de verwante onderzoeksdomeinen van de graaftheorie en de statistische mechanica.

8 Conclusies

De doelstelling van dit doctoraat was te komen tot a priori schattingen van interconnectie-eigenschappen, op basis van de interconnectietopologie, die bruikbaar zijn voor de exploratie van de ontwerpruimte tijdens het ontwerp van digitale systemen. Deze schattingen moesten in eerste instantie voldoende betrouwbaar zijn, d.w.z. dat een goede correlatie vereist was tussen de a priori geschatte en de achteraf gemeten grootheden voor individuele schakelingen.

8.1 Analyse

In het eerste gedeelte van dit werk (hoofdstukken 3 en 4) hebben we onderzocht in hoeverre de meest gangbare bestaande modellen bruikbaar zijn om het gestelde probleem op te lossen. We hebben de twee meest courante modellen, dat van Donath en dat van Davis, gevalideerd en hun voornaamste beperkingen geïdentificeerd. Bij beide bleek de regel van Rent, als benadering voor de Rentkarakteristiek, een essentiële rol te spelen. In het model van Donath heeft die een grote invloed op de betrouwbaarheid van het bereikte resultaat. In het model van Davis wordt de regel van Rent op een totaal andere manier gedefinieerd, zodat zelfs niet duidelijk is of de benodigde waarde van de Rentexponent überhaupt wel a priori te bepalen is.

We hebben dan ook een grondige analyse uitgevoerd van de Rentkarakteristiek en van de verschillende manieren waarop deze in de besproken modellen gedefinieerd werd.

De partitionerings-Rentkarakteristiek kan weliswaar a priori opgemeten worden, maar zeker voor grote schakelingen vergt dit behoorlijk wat rekentijd. Vertrekkend van een model dat werd voorgesteld door Verplaetse, zijn we erin geslaagd te komen tot goede benaderingen van dit type Rentkarakteristiek op basis van een zeer beperkt aantal parti-

tioneringsniveaus.

Vervolgens werd dieper ingegaan op de geometrische interpretatie van de Rentkarakteristiek, die ten grondslag ligt aan het predictie-model van Davis. Deze beschrijft het gemiddeld aantal pinnen van modules die overeenkomen met een gebied in een geplaatste schakeling. We zijn op zoek gegaan naar een wiskundige modellering ervan, die bovendien alle waarnemingen kan verklaren. We kunnen concluderen dat deze Rentkarakteristieken fundamenteel verschillen van partitionerings-Rentkarakteristieken. Ze zijn een gevolg van de draadlengtedistributie en worden beïnvloed door de vorm en de positie van de beschouwde gebieden. Wanneer ze worden benaderd door een machtwet (de regel van Rent), blijkt de gevonden exponent in het interval $[0.5, 1]$ te liggen voor gebieden in een 2-dimensionaal plaatsingsrooster, in tegenstelling tot de partitionerings-Rentexponent die tussen 0 en 1 ligt.

Deze bevindingen leiden ons tot het besluit dat de manier waarop de Rentexponent in Davis' model gebruikt wordt niet kan leiden tot betrouwbare a priori schattingen van de interconnectie-eigenschappen van individuele schakelingen. De methode van Donath is daarentegen wel veelbelovend.

8.2 Schatting van interconnectie-eigenschappen

Het eerste gedeelte van dit doctoraatsonderzoek heeft hoofdzakelijk geleid tot een dieper inzicht in modellen voor het a priori schatten van interconnectie-eigenschappen. In het tweede gedeelte (hoofdstukken 5 en 6) hebben we het verworven inzicht toegepast om te komen tot betere schattingen van de gemiddelde draadlengte en de draadlengtedistributie en van de klokperiode.

Eerst hebben we de methode van Donath aangepast door de Rentkarakteristiek te gebruiken in plaats van de regel van Rent. Bovendien hebben we gezorgd voor de nodige flexibiliteit met betrekking tot de architectuur en de grootte van de schakeling. Deze aanpassingen, en vooral het gebruik van de Rentkarakteristiek, hebben de betrouwbaarheid van de gemaakte voorspellingen aanzienlijk vergroot. Daarna hebben we de bezettingskans van Verplaetse aangewend om een grotere nauwkeurigheid te bereiken. Het eindresultaat zijn draadlengtedistributies die de opgemeten distributies zeer dicht benaderen. In vergelijking met één van de gebruikte plaatsingsprogramma's (Plato),

wijken de voorspelde gemiddelde draadlengtes voor bijna alle schakelingen minder dan 10% af van de gemeten waarden. Ons model blijkt vrij robuust met betrekking tot de opgemeten Rentkarakteristiek. Ook in combinatie met een geschatte Rentkarakteristiek (op basis van de modellen uit hoofdstuk 4) zijn de resultaten beduidend beter dan wat kon worden bereikt door de regel van Rent te gebruiken.

Tenslotte hebben we een model voorgesteld voor het schatten van de minimale klokperiode op basis van de draadlengtedistributie en de interconnectietopologie. Conceptueel komt dit model veel beter overeen met de werkelijkheid dan de bestaande technieken, omdat het expliciet in rekening brengt dat de klokperiode bepaald wordt door de *grootste* combinatorische padvertraging die in de schakeling optreedt. Met dit model bereiken we een grotere betrouwbaarheid, maar vooral ook een grotere nauwkeurigheid dan de traditionele schattingen, die de haalbare klokperiode sterk onderschatten. Om met dit model goede resultaten te bereiken is het echter essentieel om te vertrekken van een nauwkeurige schatting van de draadlengtedistributie.

Hoewel ons model voor de meeste geteste schakelingen vrij goed presteert, treden er voor sommige schakelingen extreem grote fouten op. We hebben kunnen aantonen dat deze te wijten zijn aan lokale afwijkingen van de homogeniteitsveronderstelling. Om deze fouten te vermijden moet dus worden gezocht naar manieren om deze lokale afwijkingen te voorspellen.

Bibliografie

- [Chr01] P. Christie. A differential equation for placement analysis. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 9(6):913–921, December 2001.
- [DBMVC01] J. Dambre, M. Brunfaut, W. Meeus, and J. Van Campenhout. Impact of optical I/O on FPGA electronic routing delays. In *Proceedings of the 27th European Conference on Optical Communication (ECOC'01)*, volume 3, pages 294–295. ACM/SIGDA, 2001.
- [Don81] W. E. Donath. Wire length distribution for placements of computer logic. *IBM J. of Research and Development*, 25:152–155, 1981.
- [DSVC02] J. Dambre, D. Stroobandt, and J. Van Campenhout. A probabilistic approach to clock cycle prediction. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 9–15. ACM/SIGDA, 2002.
- [DSVC03a] J. Dambre, D. Stroobandt, and J. Van Campenhout. Fast estimation of the partitioning Rent characteristic, using a recursive partitioning model. In *Proceedings of the International Workshop on System-Level Interconnect Prediction*, pages 45–52. ACM/SIGDA, 2003.
- [DSVC03b] J. Dambre, D. Stroobandt, and J. Van Campenhout. Improvements to Donath's hierarchical model for interconnect prediction in VLSI circuits. *submitted to IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2003.

-
- [DVSVC01] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. On Rent's rule for rectangular regions. In *Proc. of Intl. Workshop on System-Level Interconnect Prediction*, pages 49–56, March 2001.
- [DVSVC02] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. Getting more out of Donath's hierarchical model for interconnect prediction. In *Proc. of Intl. Workshop on System-Level Interconnect Prediction*, pages 9–16, April 2002.
- [DVSVC03] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. A comparison of various terminal-gate relationships for interconnect prediction in VLSI circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Special issue on System-Level Interconnect Prediction*, 11(1):24–34, 2003.
- [DVVC99a] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the impact of optical interconnect latency on the performance of optoelectronic FPGAs. In *Workshop notes of the 3rd IEEE Workshop on Signal Propagation on Interconnects*. IEEE, 1999.
- [DVVC99b] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the impact of optical interconnect latency on the performance of optoelectronic FPGAs. In *Proceedings of the Sixth International Conference on Parallel Interconnects*, pages 91–97. IEEE Computer Society, 1999.
- [DVVC00] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the performance of optoelectronic FPGAs: the impact of optical interconnect latency. *Interconnects in VLSI Design*, pages 195–202, 2000.
- [Str01] D. Stroobandt. *A priori Wire Length Estimates for Digital Design*. Kluwer Academic Publishers, April 2001.
- [VCVMDD99] J. Van Campenhout, H. Van Marck, J. Depreitere, and J. Dambre. Optoelectronic FPGAs. *IEEE Journal of Selected Topics in Quantum Electronics on Smart Photonic Components, Interconnects, and Processing*, 5(2):306–315, March/April 1999.

-
- [Ver03] P. Verplaetse. Karakterisatie van de interconnectie-topologie van digitale schakelingen en toepassingen in digitaal ontwerp, May 2003. Ph.D. thesis (in Dutch), Ghent University, Engineering Faculty.
- [VM00] H. Van Marck. Kwantitatieve studie en modellering van opto-elektronische driedimensionale interconnectiestructuren, March 2000. Ph.D. thesis (in Dutch), Ghent University, Engineering Faculty.

**Prediction of interconnect properties
for digital circuit design
and technology exploration**

Joni Dambre

Advisors: prof. dr. ir. D. Stroobandt,
prof. dr. ir. J. Van Campenhout

Contents

Frequently used symbols	iii
1 Introduction	1
1.1 The VLSI design stages, from a logic-dominated perspective	1
1.1.1 The VLSI design space	2
1.1.2 The traditional (logic-dominated) design flow	5
1.2 The technological shift towards interconnect-domination	8
1.3 Recent evolutions in VLSI design to account for interconnect cost	11
1.3.1 Interconnect-driven physical design	12
1.3.2 The construct-by-correction flow	13
1.3.3 Constructive prediction	13
1.4 Requirements for truly interconnect-driven design exploration	14
1.5 Main contributions by this Ph.D. thesis	17
2 Circuit graphs and circuit placement	21
2.1 The circuit graph	21
2.2 Rent's rule	25
2.3 The placement architecture	27
2.4 The placement process	30
2.5 Analysis of placement results	35
2.5.1 Impact of placement heuristics	35
2.5.2 Impact of the placement architecture	38
2.5.3 Impact of the interconnect topology	42
2.6 Multi-terminal nets	45
2.6.1 Length metrics for multi-terminal nets	45
2.6.2 Placement of multi-terminal nets	47

3	Fundamental approaches to interconnect prediction	49
3.1	Introduction	49
3.2	Donath's approach to estimating wire length distributions	51
3.2.1	The original model	51
3.2.2	Validation of the original model	55
3.2.3	Extensions to Donath's model	64
3.3	Geometric techniques for interconnect prediction	70
3.3.1	The first geometric wire length distribution models	70
3.3.2	Davis' concept of terminal conservation	72
3.3.3	Validation and extensions of Davis' prediction technique	75
3.4	Comparative discussion	82
4	Comparison of different terminal-gate relationships	85
4.1	Types of terminal-gate relationships	85
4.2	Circuit partitioning	92
4.2.1	The partitioning Rent characteristic	92
4.2.2	Verplaetse's recursive graph bipartitioning model	94
4.3	Geometric circuit modules	112
4.3.1	Christie's differential equation for layout Rent characteristics	112
4.3.2	Enumeration model for layout Rent characteristics	119
4.3.3	Properties of average layout Rent characteristics .	131
4.3.4	Consequences for Davis' prediction technique . .	139
4.4	Comparative discussion	143
5	Towards the accurate prediction of placement wire length distributions	145
5.1	Using the Rent characteristic in Donath's model	145
5.2	Relaxation of the architecture model	154
5.2.1	Model extensions	154
5.2.2	Validation of the extended model	157
5.3	Including better placement optimization	165
5.3.1	Stroobandt's occupation probability	165
5.3.2	Verplaetse's occupation probability	167
5.4	Sensitivity of prediction quality to the Rent characteristic	174
5.5	Chapter summary	177

6	Probabilistic prediction of the minimal clock cycle	179
6.1	The probabilistic longest path problem	179
6.2	The segment delay distribution	183
6.3	Probabilistic clock cycle prediction	184
6.3.1	Some probability theory	185
6.3.2	Equivalent independent paths	186
6.3.3	Computing the segment criticalities	187
6.4	Model validation	189
6.5	Using predicted wire length distributions	196
6.6	Conclusions	201
7	Directions for further research	203
7.1	Upwards: further model refinement and extensions . . .	203
7.1.1	Direct extensions of this work	204
7.1.2	The homogeneity assumption and multi-terminal nets	206
7.1.3	Other extensions to be considered	209
7.2	Downwards: deeper into the theoretical foundations . .	212
7.2.1	Why a partitioning based placement model works	213
7.2.2	Random graph theory	219
7.2.3	Topological properties of non-random graphs . .	221
7.2.4	Optimal bipartitioning of random and non-random graphs	226
7.3	Conclusions	231
8	Conclusions	233
8.1	Analysis and understanding	233
8.2	Improvements to the prediction of interconnect properties	235
8.3	Future work	237
A	Experimental setup	239
A.1	Benchmark circuits	239
A.1.1	Synthetic benchmarks	239
A.1.2	Public domain benchmark circuits	241
A.2	Placement tools	244
A.2.1	Recursive partitioning based placement	248
A.2.2	Flat placement by simulated annealing	253
A.3	Extraction of Rent characteristics and Rent parameters .	255
A.3.1	Measuring Rent characteristics	255
A.3.2	Fitting the Rent parameters	256

B	Basic probabilistic concepts	259
B.1	Discrete random variables	259
B.2	Moments and generating functions	261
B.3	Functions of discrete random variables	262
B.3.1	Linear combinations of independent variables . .	262
B.3.2	Absolute value of a discrete random variable . . .	264
B.4	Order statistics	264
C	Site distributions and site functions	265
C.1	Definitions	265
C.2	Site distributions for two-terminal nets	266
C.3	Site distributions for multi-terminal nets	272
C.3.1	One-dimensional length components	273
C.3.2	Coinciding gate locations	274
C.3.3	Approximation of multi-terminal wire length dis- tributions	279
C.4	Local site functions in a rectangular grid	284

Frequently used symbols

Symbol	First used	Short description
α_k	p. 95	Cut probability at partitioning level k
β	p. 102	Power law scaling exponent in the β -model expression for α_k
C	p. 33	Total placement wire length
$\mathcal{D}(\ell)$	p. 38	Site <i>distribution</i> (with index or indices if required)
G	p. 24	Number of blocks in a circuit module or its <i>size</i> (with index or indices if required)
G_c	p. 22	Number of internal blocks in a circuit netlist
\mathcal{G}	p. 112	A layout region (with index or indices if required)
H	p. 51	Number of hierarchy levels in recursive partitioning or recursive partitioning-based placement
I_c	p. 22	Number of input blocks in a circuit netlist
ℓ	p. 30	Length or distance in the layout (counted in Manhattan-units)
N		Generally used to indicate numbers of nets (with index or indices if required)
N_c	p. 24	Number of internal (two-terminal) nets in a circuit netlist
$\mathcal{N}(\ell)$	p. 33	Wire length distribution
O_c	p. 22	Number of output blocks in a circuit netlist
p	p. 25	A Rent exponent
$p_{(al)}$	p. 132	Average layout Rent exponent
$p_{(p)}$	p. 118	Partitioning Rent exponent
\mathcal{P}	p. 33	A placement process

Symbol	First used	Short description
t	p. 25	A Rent coefficient
t_G	p. 23	Average number of terminals per block
t'_G	p. 24	Average number of <i>internal</i> terminals per block
t_k	p. 23	Number of terminals connected to block k (<i>block degree</i>)
T	p. 24	Number of terminals for a circuit module (with index or indices if required)
T_{ext}	p. 24	Number of external nets in the circuit netlist
$T(G)$	p. 25	Continuous notation for a Rent characteristic (average number of terminals for a module of size G)
$T_{(p)}(G)$	p. 91	Continuous notation for a partitioning Rent characteristic
$T_{(al)}(G)$	p. 91	Continuous notation for an average layout Rent characteristic
$T_i(G_i)$	p. 26	Discrete notation for a Rent characteristic
$\mathcal{Q}(\ell)$	p. 53	Occupation probability (with index or indices if required)
$\Phi(\ell)$	p. 38	Site function (with index or indices if required)
X_g, Y_g	p. 29	Grid dimensions, counted in number of gate locations
X_c, Y_c	p. 30	Cell dimensions, counted in Manhattan units
X_l, Y_l	p. 30	Layout dimensions, counted in Manhattan units
χ	p. 88	Aspect ratio of a layout region (chosen to be $\in [0, 1]$)
χ_g	p. 30	Aspect ratio of placement grid ($= Y_g/X_g$)
χ_c	p. 30	Aspect ratio of cells ($= Y_c/X_c$)
χ_l	p. 30	Aspect ratio of layout ($= \chi_g\chi_c$)

Chapter 1

Introduction

In this chapter, we give an overview of the global context of our work which is that of VLSI design automation, including the exploration and evaluation of implementation technologies. We briefly discuss how recent technological evolutions have significantly increased the importance of interconnect and interconnect modeling in this field. Several strategies to overcome design problems due to interconnect have emerged lately. However, from a long-term perspective, fundamental solutions have so far failed to appear. Our work belongs to the research domain of system-level interconnect prediction, which targets the early prediction of interconnect related design properties. We compare the ultimate goal to the state-of-the-art in this area and identify the most urgent problems. Finally the chapter is closed by an overview of our own contributions towards the solution of these problems.

1.1 The VLSI design stages, from a logic-dominated perspective

For several decades, rapid improvements in integrated circuit technology have allowed the number of components (transistors) per chip to be doubled approximately every 18 months (Moore's law, originally formulated by Intel co-founder *Gordon Moore*, Figure 1.1). As a result, chips now contain up to several millions of transistors and the scaling still continues.

Clearly, the design of such large and complex systems cannot be achieved without an established design methodology and the support of CAD (Computer Aided Design) tools. In this section, we give an

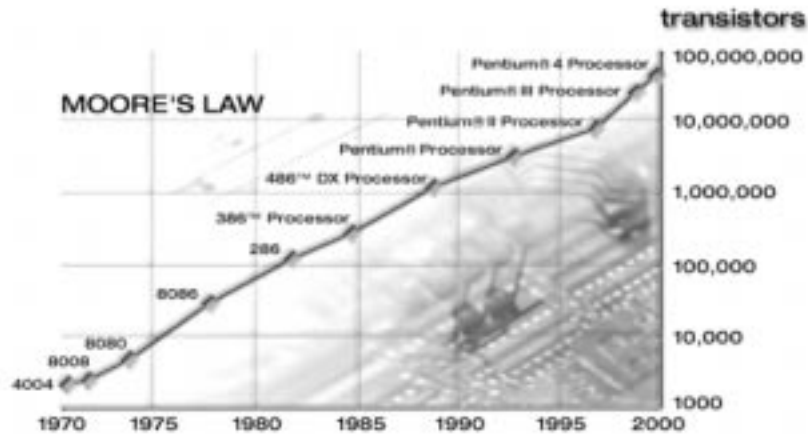


Figure 1.1: Illustration of Moore's law for Intel processors, taken from <http://www.intel.com/>.

overview of the main stages in the VLSI design flow and their combination into a design methodology.

1.1.1 The VLSI design space

Any design starts with an *informal specification* of the required design properties. This specification includes the functional behavior of the system, but also other boundary conditions, such as its timing performance, maximal power budget and reliability. Also, from an economic perspective, its maximum production cost and time-to-market are important considerations. Once the general framework has been set, the specifications are formalized as soon as possible. The economical considerations are translated into, e.g., the man-power assigned to the task and the technology that is to be used. The other specifications lead to a formal description of the systems behavioral requirements and its boundary conditions.

When the specifications have been fixed, the design space exploration can begin, with the final goal being a (digital) system that meets all constraints. Every possible solution must have the correct functionality and is evaluated with respect to (estimates of) the different cost metrics, e.g., its timing behavior (e.g., clock frequency or delay), its average and peak power consumption and its production cost (area,

1.1 The VLSI design stages, from a logic-dominated perspective 3

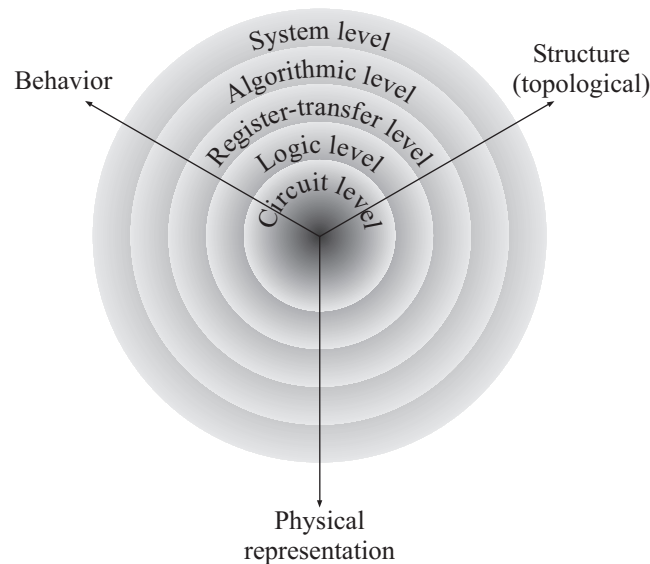


Figure 1.2: The design hierarchy levels and description views according to Gajski's Y-chart.

number of routing layers). Also, when possible, the reliability of some of these metrics is included in the overall cost picture. For instance, depending on the application, timing or functionality errors may be allowed to occur with a predefined probability.

Because of the sheer size and complexity of this design space, it is explored in several stages or *design hierarchy levels*. At each level, more detailed information regarding the costs metrics of different components becomes available and a refinement of the result from the previous level is made.

One possible representation of the design space is the Y-chart of Gajski and Kuhn [GK83] (Figure 1.2). The design can be described according to different views at each hierarchy level: its functional behavior, its topological structure and its physical representation. The exact boundaries of each hierarchy level are not always clear and sometimes differ depending on the author. In Gajski and Kuhn's Y-chart, they were described as follows:

At the **system level** the (digital) system as a whole is described. A possible behavioral representation is that of communicating processes, while the structural elements at this level can include pro-

processors, busses or large memory blocks. Physical building blocks can either be discrete components (busses, racks, boards, chips in a MCM), or regions on the chip where they are implemented (SoC or System-on-a-Chip) and interconnected;

The **algorithmic level** describes the system by way of a functional algorithm (behavior) or by, e.g., a block diagram (topological structure). Alternatively, this level is also sometimes referred to as the PMS-level (*Processor-Memory-Switch-level*). Binary signal representations need not be fixed at this level: they can still be numeric (integers, real numbers, ...). The physical description consists mainly of chip regions (e.g., large IP-blocks or memories) or, in some cases, separate physical components (e.g., dedicated accelerating processors).

At the **R/T (Register transfer) level**, the behavioral description consists of the transfers of and transformations on data. Structurally, the system can be represented as an interconnected netlist of (often combinational) low-level functionalities such as adders and multiplexers and registers. Signals can be represented as bytes or words. Physically, these building blocks correspond to small IP-blocks or standard cells, i.e. mainly regions on the chip.

At the **logic level**, behavior is represented by boolean equations and finite state machines. The structural representation of this behavior is a logic netlist, which consists of interconnected fundamental logic gates (AND, OR, XOR, etc.) and flipflops. Signals are represented as single bits at this level. Physically, the logic level components are basic library components (standard cells or custom gates), their dimensions and location on the chip and their interconnections.

The **circuit level** structure consists of a network of individual transistors. The corresponding behavioral description consists of, e.g. differential equations or Spice models, while their physical description consists of the rectangular patterns to be used as fabrication masks. At this level, signals are again continuous, now representing electronic voltages and currents.

Note that modern designs have become a lot more complex than they were when Gajski proposed his representation. Although its basic principles are still valid, the terminology and representation of

1.1 The VLSI design stages, from a logic-dominated perspective 5

the design hierarchy levels above the R/T-level have evolved in time. For example, in an alternative design space model targeting a power-optimized combined hardware/software design flow, Catthoor [Cat00] introduces the memory organization as an explicit dimension. He also introduces some additional design hierarchy levels between the algorithmic level and the final implementation (e.g., the task level, array level and process level). Linked with each of these levels is a description of the kind of design considerations and optimizations with respect to data, concurrency and memory that can be made.

In this work, we will use the terminology from Gajski's model in the context of designing the *digital computational modules* in large designs. In this context, Gajski's representation is still the most common in the EDA (*electronic design automation*) research community.

1.1.2 The traditional (logic-dominated) design flow

A design flow is the way the different levels and descriptions in the design space are traversed during the design space exploration. A typical example is shown in Figure 1.3. Hierarchically, it traverses the levels from the system level down to the circuit level.

Usually, the initial design specification describes the desired functionality of the system and fixes the global constraints that must be met. Typical behavioral constraints regard the timing properties of the system: its maximum clock frequency, the total delay (latency) from data input to data output or the minimal throughput. Other constraints are of a physical nature: the process technology and component library to be used and the maximum chip size. Also, in particular for mobile and (recently) wearable applications, the peak and average power dissipation are of increasing importance. Yet other possible constraints are the temperature range for which the system must still function, or (from an economical perspective) the size of the design team and the time-to-market. Many of these top level constraints are also refined and propagated at each refinement step of the design.

The target is a circuit-level physical description that provides the masks to be used for the chip's fabrication. During and after fabrication, ICs are also subjected to different tests. For each test, the fraction of the products (wafers, partially finished ICs or packaged ICs) that is found to meet all desired constraints is called the *yield*. Note that the minimum yield itself is also a constraint that is often included in the

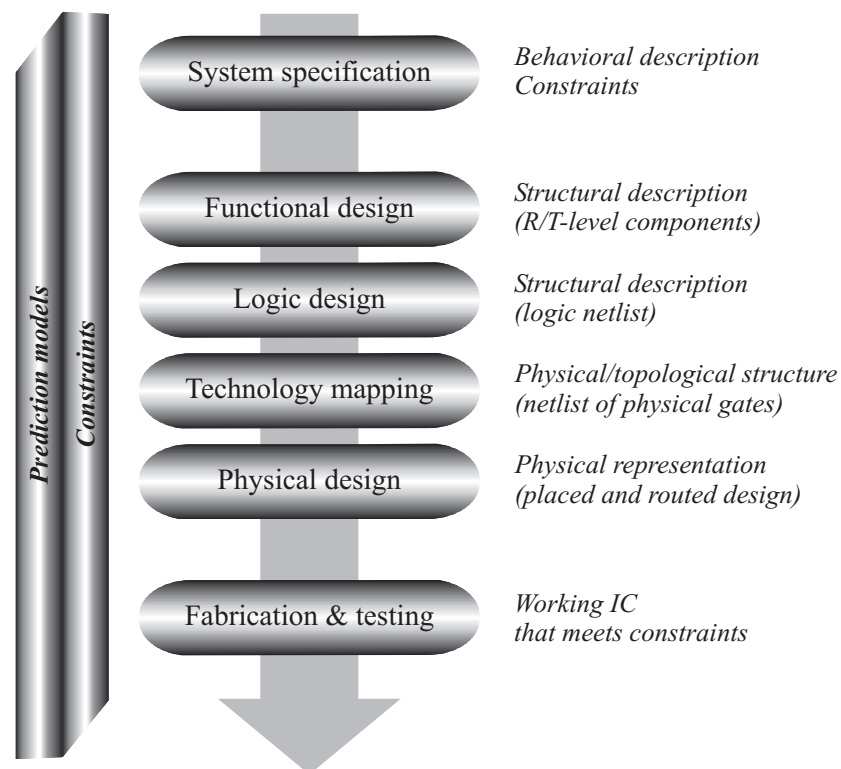


Figure 1.3: The principal stages in a VLSI design flow. Vertical arrows indicate the propagation and abstraction of cost models for increasing hierarchy levels and the propagation and refinement of constraints for decreasing levels in the design hierarchy.

1.1 The VLSI design stages, from a logic-dominated perspective 7

design specification.

Between specification and fabrication, a sequence of design steps must be traversed. The first is often referred to as **functional design**. It includes a sequence of refinements and reaches down to the R/T-level. Usually, it consists of several iterations between the behavioral and the structural domain. The structural axis is visited after each refinement step to map the behavioral description onto a partitioning of the system into smaller building blocks, which can then be described in more detail in the next iteration. The final result is a structural description at the R/T-level.

At the logic level, the functionality of the combinational R/T-level building blocks is represented by optimized boolean equations. These are then transformed by *logic synthesis* into a netlist of standard logic gates. In the **technology mapping** phase, the logic netlist is mapped onto the (often more complex) gates or cells that are available in the library (physical descriptions).¹

Although the netlist after technology mapping is already in part a physical description, the true step to the physical axis is made during the **physical design** stage. This generally consists of *floorplanning*, *placement*, *global routing* and *detailed routing*. During floorplanning, the larger building blocks in the system are assigned to regions on the chip. The subsequent stage performs the detailed placement of each gate or cell to a region on the chip. Finally, during routing, the exact path (position and routing layers) for each wire is decided, also in two stages.

The general flow direction of these steps is from behavior, over topological structure, to physical structure. Note that in state-of-the-art physical design tools, some the design steps mentioned above are not strictly uncoupled. However, in most design flows, the physical structure axis in Gajski's Y-chart is visited no sooner than it is needed to allow sufficiently accurate estimates of the different cost metrics at each level. Hence, in the traditional design flow, physical design was often kept separate from the previous design steps and was even performed by tools from a separate vendor.

The cost metric estimates that are used in each of these design steps originate from circuit level models, which are then passed on upwards through the design levels. As the level increases, abstractions are made of the models, at the cost of rapidly decreasing accuracy. As a conse-

¹In a *full custom* design flow, this stage is skipped, because all gates are newly designed.

quence, a design that meets the constraints at a given level can turn out to violate those constraints at a lower level, when more detailed information is available. When this happens, the best hope is to be able to fix the problem at that lower level. When this is not possible, it is necessary to return to a higher level and improve the design based on feedback that is generated at the lower level. Hence, instead of the simplified linear flow in Figure 1.3, iterations are often required. These can drastically increase the total design time and in some cases, convergence is not even guaranteed. Therefore, it is of the greatest importance to use cost estimations that are as accurate as possible and concentrate on optimizing those aspects of the design that have the biggest impact on each cost metric at the highest possible hierarchy level.

Our work focuses on cost metrics for the computational elements of a design (i.e., datapath and control units, as opposed to memory). For many years, most of the relevant cost metrics for these elements have been dominated by the cost of logic. For instance, the total power dissipation as well as the required chip area were largely determined by the total number of transistors or gates in the circuit. Similarly, the total delay on a combinational path between memory elements was largely proportional to the number of gates on that path (the *logic depth* of the path). Generally, the number of gates is an easily quantifiable target. It has led to several well-known optimization strategies [HS96, ADN92] that either minimize the total number of gates or the maximal logic depth. Retiming (in its original form [LS91]), a technique for timing optimization, moves flipflops across logic gates to minimize the maximal logic depth, while keeping the total number of flipflops within limits.

As a result of these logic-dominated cost metrics, the step from topological structure to physical structure could be postponed until the technology mapping and physical design stages without risking too many iterations.

1.2 The technological shift towards interconnect-domination

As technology pushes forward, different scaling trends emerge for different components. The scaling evolutions we want to focus on in this section are those for interconnects, and in particular their implications

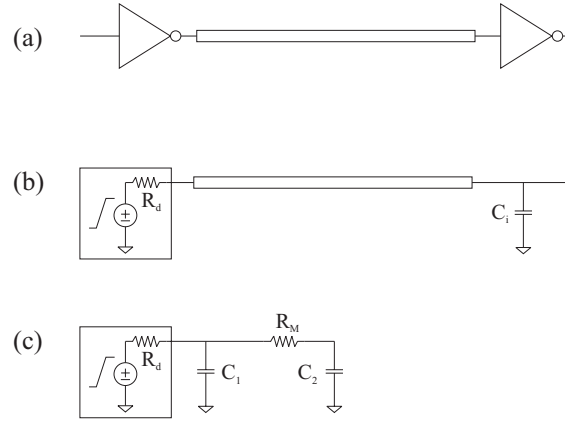


Figure 1.4: Different models for calculating the delay of a gate and the wire it drives: (a) general view; (b) using a Thevenin equivalent gate model; (c) using a pi-model approximation to capture the resistive shielding of some of the load capacitance.

for delay.

Figure 1.4(a) shows a schematic view of two logic gates, connected by a wire. In an equivalent Thevenin model (Figure 1.4 (b)), the gate output resistance R_d as well as its input capacitance C_i turn up. For a very long time, a (short) wire between two gates could be modeled as a (relatively small) capacitance and virtually no resistance. As a result, the delay of the combination of gate and wire could be modeled as a simple RC delay and was dominated by the gate model parameters.

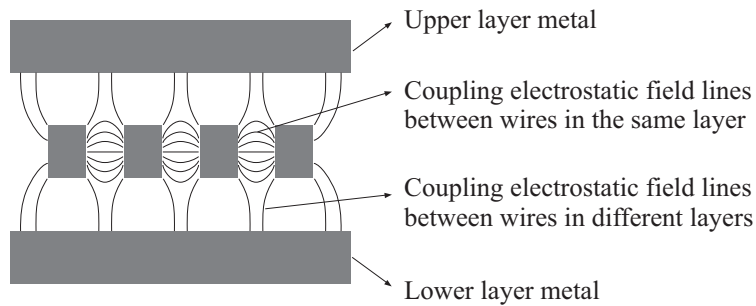


Figure 1.5: Schematic view on the physical structure of a multi-layer interconnect technology, with electromagnetic field lines indicating the origin of different coupling capacitances.

In modern DSM (*Deep SubMicron*) VLSI chips, multiple interconnect layers are present. Usually, these layers have alternating (preferred) routing directions. A schematic view on a cross-section of such a routing structure is shown in Figure 1.5. Wires are usually characterized by their (distributed) resistance r , capacitance c and inductance l per unit length. When all technology feature sizes in Figure 1.5 are scaled by a factor $S < 1$, the interconnect capacitance per unit length remains unchanged, but the resistance per unit length is increased by a factor $1/S^2$. Hence, the total RC of a wire scales as:

$$RC \sim \frac{\ell^2}{S^2}, \quad (1.1)$$

with ℓ denoting the wire length. If the gate sizes follow the same scaling trend, and if we express the length ℓ as a function of the typical gate feature size, the interconnect RC remains unchanged. However, as feature sizes decrease, the trend is to put ever more logic in a chip, resulting in equal or even increasing chip sizes. While most of the wires connect gates that are close to each other (*local wires*), there are always a number of wires that cross a large portion of the chip or a computational block on the chip (*semi-global and global wires*). As a result, the *physical* length of the longest wires remains the same or increases, and their total RC scales proportional to $1/S^2$. In reality, this analysis is somewhat simplified, because the conductor thickness does not scale with S , making r scale only by $1/S$. Also, the capacitance per unit length c does scale somewhat, resulting in an overall RC scaling factor somewhere between $1/S$ and $1/S^2$.

So how does interconnect scaling affect the total gate and interconnect delay? As features scale, so do the gate driving resistances and input capacitances. For (short) local wires, the wire resistance remains small compared to the gate resistance (both are reduced). However, because wire capacitances are not (or barely) reduced, their relative fraction in the total gate load increases.

For global wires, the distributed RC nature of interconnects becomes important. Here, the resistive component of the wires becomes comparable to or larger than the gate output resistance. Because of the distributed nature of the interconnect resistance, it *shields* some of the load capacitance from the gate output, i.e., the gate only "sees" a fraction of the total load capacitance. This can be represented by an equivalent pi-model (Figure 1.4 (c)). In delay models, the total delay is often split into the gate delay, which is obtained by assigning an *effective ca-*

capacitance as a load to the gate, and the interconnect delay. For global wires, the interconnect component of the total delay, which depends quadratically on interconnect length, represents an increasing fraction of the total delay.

Recently, because of the rapidly increasing on-chip signalling frequencies, interconnect inductive effects are also becoming important. In general, scaling trends also cause interconnect to be increasingly noisy (e.g., caused by increasing coupling capacitances). For a discussion on these effects and a general description of interconnect modeling techniques, we refer to [CPO02].

The increasing headaches caused by interconnect are addressed on different levels. On the technology level, new dielectrics and conductors (e.g., copper interconnect) are used to reduce interconnect capacitances and resistances. Though each improvement on this level causes a drop in interconnect cost, the scaling trends still continue.

On a somewhat higher level, buffers are introduced into global wires, resulting in a linear rather than quadratic scaling of delay as a function of interconnect length (e.g., [ADQ99b, ADQ99a]). This does improve the scaling trends, but since buffers consume power, this technique can only be used to a limited extent. The same argument applies to sizing the gate driving strengths (e.g., [Cou96]). As the number of interconnects that needs to be considered as global increases, power-hungry optimizations soon reach the limits of their usefulness.

So far, our discussion was focused on delay. However, as the number of gates in digital systems increases, so does the total amount of interconnect (i.e., the total length, relative to the technology feature size). To get all this interconnect routed, ever more routing layers are needed. Finally, technology scaling does not necessarily imply that height and width of the wires scale in the same way. In fact, the aspect ratio of the wires is a technological parameter that can be optimized in a trade-off between delay (dominated by rc) and power dissipation (dominated by c only) [PMC⁺03].

1.3 Recent evolutions in VLSI design to account for interconnect cost

From the previous section, we can decide that the total interconnect length as well as the length of individual wires plays an increasingly

important role in the timing behavior of VLSI designs as well as in many other cost metrics. However, in contrast to gate counts, interconnect lengths are known only after physical design. This makes it very hard for earlier design stages to perform good optimizations. In fact, when applied to new technologies, traditional optimization formulations are effectively "barking up the wrong tree" in that they are optimizing the wrong cost metrics.

Hence, it has become necessary to reconsider each design stage separately as well as the entire design flow from an interconnect-dominated perspective. One of the first driving forces was the impact of interconnect on timing. Achieving *timing closure*, i.e., the global constraint on timing and/or minimal clock cycle, has been a major headache for design engineers for over a decade. More recently, power consumption has also become an important concern, mainly for mobile applications.

1.3.1 Interconnect-driven physical design

Slowly, interconnect driven alternatives to traditional design steps have begun to emerge in the EDA literature. The first to appear were timing-driven placement and routing strategies, rapidly followed by routability-driven placement [She02, SWY03].

The main problem in timing-driven physical design is the fact that the minimal clock cycle is a global parameter, determined by the maximal delay occurring amongst all combinational paths in the circuit. In contrast, most placement and routing approaches perform local optimizations and would be slowed down immensely by having to take such a global view. The most common approach to addressing this problem is the translation of the global clock cycle constraint to delay (and hence length) budgets for each individual connection [NBHY89, SKT96, CXS00, NR98]. This is done in such a way that, if the constraints are met for all nets, the target clock cycle is also achieved. At the same time, the assigned budgets are made as large as possible, to maximize the probability that the physical design tools will indeed find a solution.

In routability-driven placement, candidate congested areas are identified during placement. This information is used to steer the placement tool away from possibly unroutable solutions, possibly at the cost of a slightly larger area or total interconnect length [BR97,

WYES00, LKS01, YKS02].

1.3.2 The construct-by-correction flow

If, in a timing-driven physical design flow, timing violations do occur for individual wires, this does not necessarily imply that timing closure is not achieved, because other connections may turn out much faster than their assigned budgets. Still, small or larger iterations are often required. If, after performing the necessary updates, placement and routing are performed all anew, there is a very big risk of causing a lot of new constraint violations and convergence may be very slow.

To try and prevent this, incremental physical design tools emerged next. These construct a (partial) physical design in such a way that it is easy to make incremental improvements to the netlist. Now, each local improvement or group of improvements can be followed by locally updating the placement and routing, while maintaining as much of the previous solution as possible. This strategy effectively minimizes the risk of creating new problems while solving the old ones.

Examples of small local improvements are the insertion of some additional buffers or the increase of the driving capacity of certain gates to speed up the slowest connections. Somewhat larger changes include retiming (e.g., [IK01]) or limited logic restructuring (e.g., [GKSV01, SB02]).

A survey of incremental CAD techniques is given in [CCMS00].

1.3.3 Constructive prediction

Despite all efforts, it is possible that the budgets and constraints, generated at the logic and R/T-levels are simply not realizable. When the constraints in a particular solution can not be met, taking it all the way through physical design stages is definitely a waste of time. It would be far better to exclude it in advance. Also, there usually exists a wide range of possible constraint sets for individual connections that all satisfy the global clock cycle constraint. To maximize the chances of success, one should be able to pick those constraints that make the task of the physical design tools as easy as possible, by assigning large budgets to wires that have a large probability of ending up long. As an additional benefit, this may reduce the need for power-hungry interventions (such as buffer insertion).

However, to achieve this, advance information on (probable) individual wire lengths after placement is required. One way to provide such information is by making a *constructive estimate* of the wire lengths. This is done by performing a quick-and-dirty placement or floorplanning of the design and using the resulting lengths as an estimate (e.g., [KS01]).

Alternatively, part of the physical design is pushed upward in the design hierarchy. For instance, Yang et al. [YCS02] propose to produce a preliminary floorplan as early as possible (at or above the R/T-level). When a floorplan for the larger design modules is available, more accurate estimates can be made for (at least) the long interconnections. After designing the individual modules, the remainder of the physical design is done, while maintaining as much of the original floorplan as possible. In [CL00], an interconnect-centric design flow for integrated retiming and floorplanning is proposed. Their approach is based on simultaneous recursive partitioning, placement and retiming, while minimizing the cut size at each level (routability) and taking the impact on timing into account. This approach was integrated in an entirely constructive interconnect-centric design flow [Con01].

1.4 Requirements for truly interconnect-driven design exploration

So far, the techniques mentioned above have succeeded in shortening the design cycle, first by providing fast local iterations to fix constraint violations and then by trying to prevent them by constructing solutions while taking interconnect lengths into account. As such they answer to urgent cries from the digital IC industry. In the 2001 edition of the ITRS (*International Technology Roadmap for Semiconductors*, [ITR01]), the message was given that the cost of design, to a large extent determined by the total design time and man power, is rapidly increasing, relative to the production cost.

However, when evaluating the previous approaches from a medium to long-term perspective, we must keep in mind the key goal of efficient design, namely to optimize as much as possible at the highest possible level of abstraction. To allow this, we need to produce better predictors for interconnect related cost and new strategies to optimize those costs. Only in this way, we will ever be able to optimize the right

cost function at all hierarchy levels.

While constructive prediction can be used to generate better constraints for downstream design steps, they can not be abstracted and propagated upwards in the design hierarchy. Also, though they do make certain information available at an earlier stage than was originally the case, they still require a significant computational effort. This makes them rather unsuitable for a wide-range design space exploration. This is because they do not truly make the link between the *interconnect topology* of the circuit at a given level and the optimization job the physical design tools will make of it.

What is really needed are prediction metrics for design space exploration and optimization. These are to replace the traditional metrics, which are mainly dominated by gate count, to guarantee that *the right* costs are minimized. Because they are to be used in cost functions of minimization formulations, they need not be extremely accurate, but they must show a good correlation with the final result. After the design space exploration has come up with a single or a few well-optimized solutions, the aforementioned constructive and iterative techniques can still be used to guarantee that all constraints are met. However, a good starting point will definitely increase their chances of success.

Over the past few years, some such metrics have started to appear for individual design steps. Some use a much simplified model for placement, without actually performing it (e.g., [GNBSV98]), while others actually identify and optimize metrics, extracted from the circuit topology, that appear to correlate well with either routability or timing (e.g., [KSD02, UKK02, KK03, YM03]).

A different strategy altogether is to make sure that the design quality is predictable by using a technology that leads to very regular and predictable implementations. Examples are the combinational multi-level River PLAs [MB02a] and sequential Whirlpool PLAs [MB02b].

The study of the link between the circuit interconnect topology and the cost of its physical implementation has been the main subject of a *priori* or *system level interconnect prediction* (SLIP) since the 1960's (introduction of Rent's rule). In 1999, the networking between several researchers in this domain resulted in the first *Workshop on System Level Interconnect Prediction* [SLI99], which has since then become a yearly event.

The earliest contributions to SLIP research provided models for the

average wire length and the distribution of wire lengths in a circuit implementation [Don79, Feu82, SP86, DDM98, Str01b]. Essentially, these predictions combine models for the *circuit topology*, the *physical implementation substrate* and the strategy or the results of *embedding* that circuit in the given substrate [Str01b]. The model for the implementation substrate also includes the average physical properties of the basic building block implementations (library cells) and is often referred to as the implementation *architecture*.

The final goal of SLIP-techniques is to provide predicted wire lengths and models for interconnect delay, area or routability and to combine them into cost predictions for design space exploration at different abstraction levels. Depending on the level of abstraction on which they are to be used, more or less information is available for prediction. Usually, predictors from the higher levels are derived from more detailed, lower level predictors, for instance by replacing detailed information by typical values. As the prediction quality necessarily decreases for higher abstraction levels, it is important to have low-level predictors that are as accurate as possible. This is why the focus of interconnect prediction research should be on the lowest levels first.

Hence, in the ideal situation, a priori interconnect prediction should be able to produce (non-constructive) logic or R/T-level wire lengths for individual connections. Several predictors for length-related cost as a function of wire length [Sak93, CP99] can subsequently be applied. The resulting length-related costs should be reliable, i.e., they should correlate well with the final cost of each individual wire, not only across different wires in the same design implementation, but also across implementations in different layout substrates (e.g., different materials, buffering strategies, wiring schemes, cell aspect ratios, ...) and across different topological alternatives.

The reality of state-of-the-art interconnect prediction is still far removed from this ideal picture. Since the emergence of the corner stone of most current prediction techniques, Rent's rule [LR71a], the focus has been mainly on its application. Grossly simplified, Rent's rule states that a power-law relationship exists between the number of gates in a portion of the design (a *module*) and the number of wires needed to connect that module with the rest of the design or its exterior. However, as we will show in this work, predictions based on this power law are not sufficiently accurate to obtain the desired correlation across different circuit topologies.

This is one of the main reasons why, although some of the standard prediction techniques date from several decades ago, the majority of applications still lie in the area of *technology evaluation*. These try to assess the impact of technological evolutions or variations in the geometry of the layout substrate for *typical* circuits from a given category. Well known examples are the *International Technology Roadmap for Semiconductors* [ITR01], the *Berkely Advanced Chip Performance Calculator* (BACPAC, [SK99]), the *Marco GSRC Technology Extrapolation (GTX) System* [CCK⁺00] and evaluations of the benefits of using (optical) 3-D interconnect, or 3-D integration (e.g., [VM00, MY87, SSBK00]). However, in most of these applications, *typical values*, meant to characterize classes of applications, are used for the parameters in the circuit model. In [Str01b], Stroobandt explicitly states that:

“I do not intend to provide estimates for one specific circuit or a specific architecture. On the contrary, I want to estimate properties for collections of different circuits and different architectures, characterized by certain properties.”

Another factor that prevents real breakthroughs of SLIP techniques for design space exploration is the fact that they succeed only in predicting the overall wire length distribution. In contrast, most of the design problems today are due to a few global connections. Therefore, to be really effective, predictions of individual wire lengths should be made.

The few non-constructive predictions that do target specific circuits or individual wires [BN01, YM03] concentrate on very local metrics and optimizations during logic restructuring or physical design. Some of the references mentioned in the previous section fall into this category. However, many of those lack theoretical foundations and some are even entirely empirical.

1.5 Main contributions by this Ph.D. thesis

From the discussion in the previous sections, it is clear that many problems remain to be solved in the area of system-level interconnect prediction. More specifically, some of the most urgent problems are the bad correlation with individual circuit results, the modeling of wires that connect more than two gates (*multi-terminal nets*, as opposed to

two-terminal nets), the prediction of individual wire lengths and the inclusion of routing and routing congestion.² It is our opinion that these problems should be addressed essentially in the aforementioned order.

In the long run, the target of solving these problems is to guide automated interconnect optimization earlier in the design cycle. However, in the short run, any improvement to the reliability of interconnect prediction techniques is immediately applicable for technology exploration.

In this work, we start by addressing the first problem:

the prediction of wire length distributions and average wire lengths for specific circuit topologies (with two-terminal nets only) and physical architectures.

Our approach is to first evaluate the two most important existing prediction techniques (Donath's technique [Don79] and Davis' technique [DDM98]) and to pinpoint their main limitations and causes of inaccuracy (Chapter 3). It turns out that most of the existing inaccuracy is caused by assuming the power-law relationship of Rent's rule. However, both techniques also offer only a limited capability or accuracy in modeling architectural variations. Because both prediction approaches rely strongly on different interpretations of Rent's rule, we go in search of better models to replace it (Chapter 4). We conclude that predictions based on Donath's approach seem the most promising to achieve the necessary correlation with the topology of individual circuits as well as parameters of the layout substrate.

In Chapter 5, we provide the necessary extensions to Donath's model. The result requires somewhat more computation than the original model, but it offers a very high correlation with placement results from three different tools, across different circuit topologies as well as a high correlation across different architectural alternatives.

To predict interconnect-related cost metrics in digital designs, the accurate prediction of the wire length distribution is only a first step. In Chapter 6, we address the prediction of one of the most complex derived metrics: the achievable minimal clock cycle. In the prediction of this metric, the interconnect-related delay is rapidly becoming the dominating factor. We show that this evolution necessitates the modeling of

²In most of the theoretical work on interconnect prediction, the predictions model and are compared to placement results only, because they do not include the impact of routing and routing congestion.

the achievable clock cycle in a probabilistic way. In particular, the fact that the minimal clock cycle is determined by the *largest* delay, occurring amongst all combinational paths in the circuit, needs to be modeled explicitly. The proposed prediction model uses the predicted wire length distribution and interconnect delay models. Additionally, some other parameters are extracted from the circuit topology, to capture the combined effect of a large number of combinational paths. Compared to existing approaches, the clock cycles predicted by our model are more accurate and correlate better with experimental results.

This work is concluded by an overview of topics for further research and some suggestions on how to tackle them. Although a lot of work remains to be done, it is our belief that the increased understanding as well as the measurable progress achieved in this Ph.D thesis will help solving many of the remaining problems.

The most important results from our work have been presented at international conferences and (in part) published in or submitted to international journals. The more theoretical work on circuit modules described in Chapter 4 can be found in [DSVC03a] and [DVSVC01, DVSVC03]. The analysis of and extensions to Donath's interconnect prediction model are the subject of [DVSVC02, DSVC03b], and the model for clock cycle prediction was published in [DSVC02].

Our work on the prediction of wire length distribution was rooted in the experience that was already present in our research group (mainly the work of Herwig Van Marck [VM00] and Dirk Stroobandt [Str01b]). More recently, the discussions and exchange of ideas with Peter Verplaetse were very interesting and have contributed to this work. In his Ph.D. research, he concentrated on studying and modeling the topological properties of circuit graphs. For the work on clock cycle prediction we have greatly benefited from previous, mainly empirical research on optical inter-chip interconnect in FPGAs and its impact on their performance, e.g., [DVVC99a, DVVC99b, VCVMD99, DVVC00, DBMVC01].

Chapter 2

Circuit graphs and circuit placement

In this chapter, we first introduce the circuit graph model used in the rest of this work (Section 2.1). One of the main ideas behind interconnect prediction is to extract from the circuit's interconnect topology, represented by the circuit graph, a limited number of global parameters that are best suited to model the interconnect-related properties of the circuit as a whole after layout optimization. In this light, we discuss Rent's rule, a long-standing model for describing circuit interconnect topologies, in Section 2.2. A simplification that is fundamental to most interconnect prediction techniques is the fact that circuit placement is used as a model for the entire layout process. This implies that assumptions are made with respect to the layout substrate (or placement architecture) as well as the actual placement process. These models are described in Sections 2.3 and 2.4. Then, in Section 2.5, some experimental results are shown to indicate the impact of the circuit topology, the placement architecture and the placement process on the final placement quality. We end this chapter with a brief discussion of the problems associated with predicting the lengths of multi-terminal nets.

2.1 The circuit graph

Interconnect prediction techniques usually take a structural view on *circuits*. They are described as a *netlist*, i.e., a set of functional blocks, with interconnections between them. When interconnect prediction was first introduced in the late 1970's [Don79], this description was mainly on

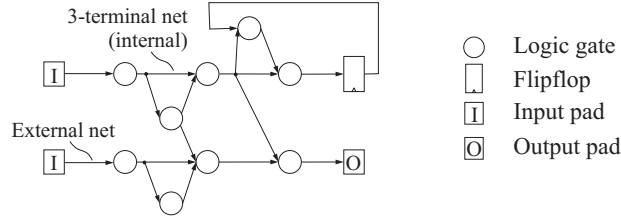


Figure 2.1: Model for the circuit topology with indication of different elements.

the logic level, with the blocks representing elementary logic gates and flipflops and the signals corresponding to a single bit. However, as designs grew larger, so did the basic building blocks. Currently, the basic elements are typically standard cells from a cell library, but some of them may just as well be large IP-blocks, with byte or word-width signals exchanged between them. As such, the exact hierarchy level of the netlist is not well-defined. In this work, we will assume that the mixture of sizes of the basic elements is not too heterogeneous.¹

The level of detail that needs to be incorporated in a circuit model depends largely on the prediction application. When only wire lengths need to be predicted, the netlist can be strictly structural, making abstraction of the behavior of the functional blocks, as well as signal direction. However, because we also want to predict timing properties such as the maximal clock cycle (Chapter 6), we need to make a distinction between combinational blocks and sequential blocks or flipflops and between block inputs and outputs.

The most detailed circuit model used in this work (figure 2.1) is a directed graph. It consists of a set of elementary blocks (or simply *blocks*) and a set of directed interconnections or *nets*. Dedicated high-fanout nets such as clock nets are not included, because they are often treated separately during physical design.

Blocks

The elementary blocks consist of G_c *gates* or functional units of approximately equal size and dimensions, as well as $\{I_c, O_c\}$ input pads and output pads, respectively, to provide communication with the circuit's exterior. To model synchronous behavior (Chapter 6), we also make

¹A first exploration of the impact of different block sizes was performed in [Ver03].

a distinction between *synchronous* or *clocked* block outputs (i.e., block outputs on which transitions are synchronous with a global clock signal) and *asynchronous* or *unclocked* block outputs. The most elementary blocks with clocked outputs are flipflops (Figure 2.1). They are also the synchronous blocks that typically occur in logic netlists. However, more complex synchronous gates can occur. We assume that all signals arriving at input pads are clocked and that flipflops are used to buffer the output signals.

Nets and terminals

In real circuit netlists, the connections or *nets* each connect two or more blocks. *Terminals* are the connections between a net and a block. Each net has one driving terminal or *source* and one or more output terminals or *sinks*.² Nets with only one sink are called *two-terminal nets* while the others are referred to as *multi-terminal nets*. The *net degree* is the total number of terminals on a net, equal to its number of sinks plus one. Similarly, the *block degree* t_k of a block g_k is the total number of terminals on a block, i.e., the sum of its number of inputs and its number of outputs. We use the symbol t_G to denote the *average block degree* for gates or the *average number of terminals per gate*.

The effective modeling of multi-terminal nets in interconnect prediction is a difficult problem for which only partial solutions exist, e.g., [Str01a, Str01b]. We consider this problem to be somewhere between modeling placement and modeling routing (see Section 2.6). When we started this thesis research, placement modeling was still very incomplete and routing models were almost nonexistent. It was our intention to study the problem of placement interconnect prediction without multi-terminal nets more thoroughly, because we believe that it captures the essence of interconnect prediction. As such, a deeper understanding of this problem is a prerequisite for addressing more complex issues such as routing, but also routability driven placement and timing optimization. Hence, in this work, we use a graph instead of a hypergraph to model circuit netlists. This implies that a one-to-one relationship between graph edges and signals can no longer be assumed. Instead, the presence of an edge between two nodes represents that there is a directed (source-to-sink) connection between both. The multi-terminal nets from the original netlist are represented by

²We do not include busses or bidirectional input/output pads in our models.

multiple edges, with the same source node as the original net, shifting all fanout from the nets to the graph nodes.

In this graph representation, nets that are connected to at least one input or output pad are called *external nets*, while all others are *internal nets*. The number of internal nets in the graph is denoted as N_c . In some cases, when the number of external nets is very large, they can strongly influence the global wire length distribution. In some approaches to interconnect prediction, they are modeled separately and then added to the global wire length distribution [SVMVC97, Str01b]. However, the wire length prediction techniques mentioned in the rest of this Ph.D. thesis focus on internal nets and internal terminals. We expect this to be a reasonable approximation as long as the number of external nets is only a very small fraction of the total number of nets. External nets are only used in Chapter 6.

Terminals that connect a block to an external net are called an *external terminal*, while all others are *internal terminals*. The total number of external terminals in the circuit will be referred to as T_{ext} . On average, the number of external terminals per gate equals T_{ext}/G_c . We use the symbol t'_G to indicate the average number of *internal terminals* per gate. It is given by:

$$t'_G = t_G - \frac{T_{ext}}{G_c} \quad (2.1)$$

Modules

A circuit *module* is defined as any subset of the circuit's internal blocks. The module size G is the number of internal blocks in the module, and its terminal count T is the number of nets connecting blocks inside the module with blocks outside the module or with the circuit's exterior. Hence, for a module that contains all of the blocks, $T = T_{ext}$. The nets, connected to those terminals are called external nets for the module.

Experiments

Throughout this work, we have performed a large number of experiments on different series of benchmark circuits. Some of those (MCNC [CBL], ISPD98 [Alp98]) are standard, publicly available benchmark series that are widely used to report research results in the fields of, e.g., design automation or interconnect prediction. Others are synthetic benchmarks that were generated by `gn1` [SVVC00] and have

controlled circuit properties. Those benchmark series and their most important properties are described in Appendix A.

2.2 Rent's rule

The early beginnings of the current interconnect prediction techniques lie with circuit partitioning. One of the first approaches to computer aided circuit placement, and one that is still very popular today as a first step, is based on recursive circuit partitioning. At each partitioning step, the number of nets cut or a similar cost function is minimized, while enforcing the resulting subcircuits (*modules*) to have close to equal size (*balanced partitioning*). In the 1960's, E.F. Rent, an IBM employee, noticed that the average number of terminals required to connect the gates in a module to the rest of the circuit and its exterior approximately follows a power law as a function of the number of gates in the module. He published his findings in two IBM internal memoranda, which led to the first publication regarding such a *terminal-gate relationship* by Landman and Russo in 1971 [LR71b]. They hierarchically partitioned the circuit into modules, at each hierarchy level minimizing the number of terminals, while keeping the variance on the module sizes as low as possible. For these partitionings, they plotted the average terminal count versus the average module size to which they fitted a power law

$$T = tG^p. \quad (2.2)$$

Originally, Rent's rule was defined for modules resulting from recursive circuit partitioning. However, there exist many strategies for generating circuit modules. For instance, they can also be created by some bottom-up clustering approach. A totally different approach is to define modules that consist of the gates that are located within the boundaries of a given region in a circuit layout (see Chapters 3 and 4). Various interpretations how modules are to be generated, and of the resulting power-law terminal-gate relationship now known as *Rent's rule*, have been the basis for most interconnect prediction techniques we know today [CS00] (see also Chapter 3). The coefficient t and exponent p are now generally referred to as the *Rent coefficient* and the *Rent exponent*, respectively.

Rent's rule (Equation 2.2) is an approximation for the *Rent characteristic*³ $T(G)$, which expresses the *average* terminal count T for mod-

³This terminology was first used by Verplaetse in [VDSVC01]

ules of a given size G . As we will discuss in Chapter 4, many different module generation strategies result in an approximate power-law terminal-gate relationship. Disregarding the way it was generated, the Rent characteristic can be described as a set of measured data points $T_i(G_i)$ for which the G_i are more or less equally spaced in the logarithmic domain. However, we will often use the continuous notation $T(G)$, implicitly assuming that T -values that are not available can be estimated by interpolation in the logarithmic domain.

If Rent's rule were to reflect reality exactly, the extreme values for G ($G = 1$ and $G = G_c$, respectively), would result in the average number of terminals per gate

$$t_G = t(1)^p = t. \quad (2.3)$$

and the number of external nets in the circuit T_{ext} :

$$T_{ext} = tG_c^p. \quad (2.4)$$

These two boundary conditions then directly yield the corresponding Rent parameter values:

$$\begin{aligned} t &= t_G \\ p &= \frac{\ln(T_{ext}) - \ln(t_G)}{\ln(G_c)} \end{aligned}$$

Unfortunately, though Rent's rule offers a relatively accurate approximation for a broad range of module sizes, it usually deviates from the Rent characteristic for both large and small modules. In the past, the region where Rent's rule offers a good approximation was called *region I* of Rent's rule. The regions where deviations occur were referred to as *region II* of Rent's rule [LR71b] for large module sizes, and *region III* of Rent's rule [Str99] for small module sizes.⁴ Therefore, if it is available, using the Rent characteristic $T(G)$, rather than a simplified expression derived from it (such as Rent's rule) can be expected to yield more accurate results.

If Rent's rule is used in interconnect prediction, a two-parameter power-law must be fitted to obtain values for t and p that minimize the prediction error. Figure 2.2 shows a plot of the number of terminals

⁴Originally, regions II and III were defined as regions of partitioning-based Rent characteristics. However, we will extend this terminology to all Rent characteristics, disregarding the way the modules were generated.

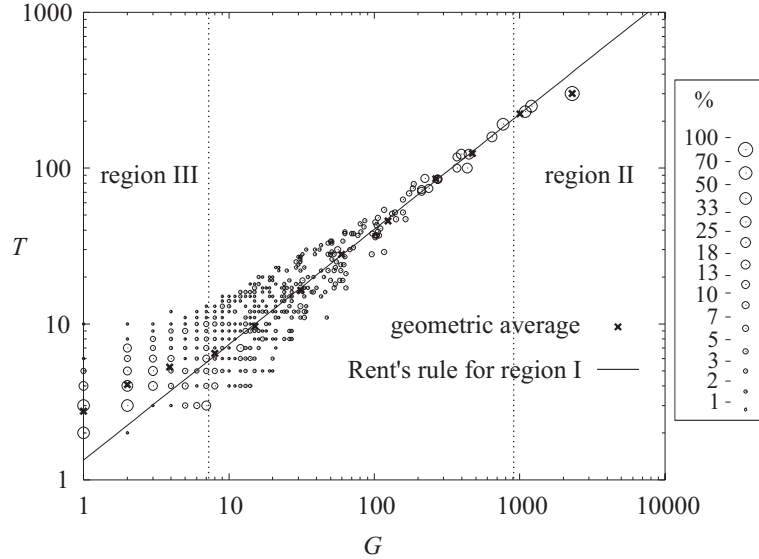


Figure 2.2: Typical example of the terminal-gate relationship for a hierarchical bi-partitioning of a circuit (ISCAS85 benchmark `c5315nr`), showing geometric average values of T vs. G and the optimal, region I power-law fit.

as a function of the module size for all modules in a hierarchical partitioning of ISCAS85 benchmark circuit `c5315nr`, as well as the resulting Rent characteristic and a power-law fit. Depending on the specific application, the strategy for Rent's rule fitting may differ, but generally the boundary conditions (2.3) and (2.4) no longer hold. The fitting algorithm used in this work is described in Appendix A. Table 2.1 shows the values of t and p resulting from a power-law fit to region I for the partitioning Rent characteristics of benchmarks `ibm01-ibm18`. In Chapter 4, we will explore models for different types of Rent characteristics that provide a better fit than Rent's rule and for which conditions (2.3) and (2.4) are satisfied.

2.3 The placement architecture

Circuit placement consists of assigning a unique physical position to each circuit gate. These positions must be arranged on one or multiple physical planes, such that within each plane, no two gates are overlapping. Usually, there are certain rules regarding the minimal separation of the gate positions. In real circuit designs, gates can have very differ-

Table 2.1: Values of t and p resulting from a region-I fit for recursive bipartitioning Rent characteristics (two-terminal net versions of benchmarks ibm01-ibm18).

benchmark	t_{fit}	p_{fit}
ibm01	11.26	0.531
ibm02	8.57	0.713
ibm03	7.69	0.675
ibm04	6.44	0.689
ibm05	7.80	0.755
ibm06	8.46	0.673
ibm07	9.58	0.622
ibm08	9.00	0.677
ibm09	12.91	0.563
ibm10	11.95	0.609
ibm11	9.73	0.608
ibm12	12.46	0.644
ibm13	11.21	0.620
ibm14	7.85	0.668
ibm15	10.73	0.581
ibm16	11.24	0.637
ibm17	17.36	0.615
ibm18	11.96	0.615

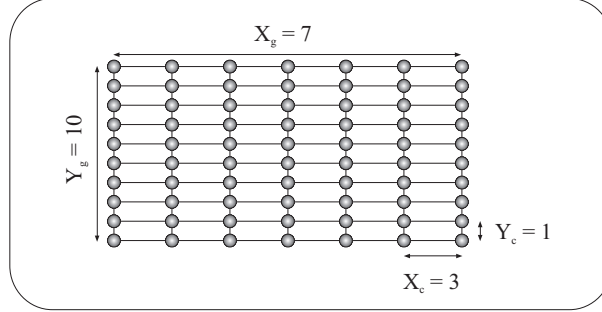


Figure 2.3: Illustration of the different parameters in a (two-dimensional) placement architecture.

ent sizes, including, e.g., large IP and memory blocks.

Depending on the design strategy and technology, there can be further restrictions on the gate positions. In standard cell designs, the gates (library cells) have a fixed height and are more or less arranged into equally spaced rows. Also, there might be a global constraint that dictates or restricts the overall size and shape of the total layout area, or certain parts of the layout region can be *forbidden regions*, reserved for other parts of the system. Such constraints will not be incorporated in our models.

In interconnect prediction techniques, it is common practice to approximate any continuous range of allowable gate positions by a discretized one, i.e., a regular two-dimensional (or three-dimensional) *placement grid*. In this approximation, each grid location is allowed to contain a single gate only and every gate is assumed to fit within a single grid location. For all cases where the individual gate sizes do not differ too much and in sufficiently large designs, we can expect that the errors introduced by this discretization are largely averaged out.

The parameters of the placement grid, i.e., the metrical physical aspects of both the circuit and the implementation technology, are combined into an *architecture* or *layout medium* model (see Figure 2.3). It comprises possible gate and pad positions, gate pitches and gate aspect ratios. The *grid dimensions*, i.e., the *number of grid positions* in each direction, will be referred to as X_g and Y_g . Although all of our models can be extended to three-dimensional systems, we only consider planar chips in this thesis.

We allow our architectural model to have a different unit distance

for each direction, expressing, for instance, rectangular cell shapes (when using standard cells). Though we will regard these dimensions as lengths in this thesis, they can also be used to model different length-related interconnect costs in each direction [VM00]. Disregarding the underlying physical reality, we refer to these unit distances as the *cell dimensions* X_c and Y_c . The *layout dimensions* $X_l = X_g X_c$ and $Y_l = Y_g Y_c$ can be derived from these parameters. In some cases, we use the aspect ratio χ , rather than the actual dimensions, as a parameter. Again, the distinction must be made between *grid aspect ratio* ($\chi_g = Y_g/X_g$), *cell aspect ratio* ($\chi_c = Y_c/X_c$) and *layout aspect ratio* ($\chi_l = \chi_g \chi_c$).

Throughout this Ph.D. thesis, we will rarely consider external nets and I/O-pads. However, when we do consider them (in Chapter 6), we assume the I/O-pad locations to be in a ring around the cell placement grid. Alternatively, they can be assumed to be on a different layer, spread out evenly across the cell placement area (*area-I/O* instead of *perimeter-I/O*, [VM00]). In both cases the pads can have a different pitch than the cells.

In this architecture, distance is defined using a Manhattan (or L_1) distance metric, i.e. the distance between two grid locations i and j , with grid coordinates (x_i, y_i) and (x_j, y_j) , respectively, is given by:

$$d_{i,j} = |x_i - x_j| + |y_i - y_j|. \quad (2.5)$$

The post-placement *length* ℓ of a two-terminal net equals the distance between the grid locations that are occupied by the gates this net connects. This is also the minimal routing length for that net, under the assumption that all net terminals are placed at the center of a grid location.

2.4 The placement process

The easiest way to perform a *valid* placement, using the placement architecture defined in the previous section, is to randomly assign each gate to a grid location, disallowing overlap. However, in real circuit design, the target is to perform a *good* placement. Generally, such a placement is characterized by several metrics. Some of these metrics must be optimized, while others are required to meet certain constraints. Some of the final metrics by which a design is evaluated can be: the interconnect-related part of total power dissipation (usually budgeted), the achievable clock cycle (also constrained), or the routability and total

routing cost (often restricted by a maximum number of routing layers). The most important interconnect properties that are affected by circuit placement are (in order of increasing detail):

- (i) the total interconnect length,
- (ii) the number of nets of each length,
- (iii) the lengths of individual wires and
- (iv) the regions through which these wires must be routed.

Although the overall quality of a placement can be determined by a very complex combination of different cost metrics, it is usually improved by the minimization of total interconnect length. Indeed, if the total wire length is reduced, so are the probabilities of individual wires being long. Overall, long wires cause increased power consumption (caused by the repeated loading and unloading of the wire capacitance) as well as long delays. Hence, optimizing the total interconnect length usually has a positive effect on the achievable clock cycle. Also, as individual (post-placement) interconnect lengths are shorter, the regions through which they have to be routed are smaller. This reduces the probability that (too) many wires need to be routed through the same layout region (i.e., *routing congestion*). Hence, the total wire length can always be considered as a (simplified) first order quality metric, even with respect to more complex metrics such as clock cycle and routing requirements.

However, both clock cycle and routing congestion often depend on so called *hot spots* in the circuit layout, i.e., the cumulative effect of a relatively large number of random variables (wire lengths, routing regions, ...), all having *bad* values. In some cases, such hot spots can be partially identified from the circuit topology. For instance, in many circuit designs, some wires have a larger impact on the achievable clock cycle than others. Hence, to effectively optimize the clock cycle, some wire lengths need to be very short, while others are less critical (see Chapter 6). Also, certain portions of a circuit may have a high probability of causing routing congestion, while others have very low routing requirements. Again, the optimization of routability should incorporate these effects. Unfortunately, in most cases the exact location and the impact of these hot spots can not be identified a priori. Often, these properties are very hard to predict even when placement information is available [SN00]. In those cases, performing a (partial) circuit routing

is the only way to accurately evaluate placement results. Obviously, such a procedure would be far too slow to be applied in the inner loops of placement optimization. Hence, even in state-of-the-art placement tools, the basis of the optimized cost function still remains the total wire length. The optimization of timing or routability is often included by assigning weights to the individual wires, which are sometimes allowed to change during the placement process. The specific cost function that is targeted by the EDA tool flow can have a significant impact on the final result. Furthermore, some optimization criteria tend to work against each other, so the physical implementation is always the result of some trade-offs.

If realistic metrics are hard to predict when (preliminary) detailed placement is available, it is even harder to predict the result of a placement process that optimizes them with some accuracy. However, achieving high accuracy is not the main purpose of this work. Our aim is to identify prediction models that are good for design space exploration. These models should predict the *optimizability* of the interconnect properties for a given design topology. Instead of accuracy, we mainly require a high correlation between predicted and actual results. This guarantees that the predictions are sufficiently *reliable* to be the basis for making trade-offs. It can be hoped that, to some extent, such predictions can also express the optimizability under different interconnect-related cost functions such as routability or clock cycle minimization.

One of the basic principles of interconnect prediction is to make abstraction of local information as much as possible. Instead, global parameters are extracted (e.g., the Rent characteristic). Because local information is no longer available, it is implicitly assumed that all wires are *statistically equal*. Hence, in most interconnect prediction techniques, the placement problem is implicitly defined as follows:

Assign a unique grid position to every circuit gate, such that the sum C of all wire lengths resulting from that assignment is minimized.

Clearly, this placement problem is NP-hard, with a solution space of (for the 2-D case)

$$\frac{(X_g Y_g)!}{(X_g Y_g - G_c)!} \quad (2.6)$$

different solutions to explore. Hence, for non-trivial placement problems, it is not feasible to try to obtain a placement for which C equals its absolute minimal value. Instead, the placement problem is generally solved by heuristics that use a certain degree of randomization to explore the solution space. Obviously, making different randomized choices results in different final placements and hence different individual wire lengths.

Across this set of optimized placement solutions, the placement process has a stochastic nature, and the achieved total wire length C as well as the lengths of individual wires become random variables. What we aim to predict in this work are the *expected interconnect properties* of the so defined stochastic placement process.

A placement process \mathcal{P} that directly minimizes the total wire length C as a cost function can be called *homogeneous*, because the cost of all individual wires is equal to their length at all times during placement. Under the assumption that all wires in the circuit topology are statistically equal, this implies that

the post-placement lengths of the individual (two-terminal) wires are equally distributed random variables.

This last assumption will be referred to in this work as *the homogeneity assumption*. The probability mass function of the post-placement wire lengths is generally referred to as the *wire length distribution* $\mathcal{N}(\ell)$.⁵ Because of the restrictions we made on the placement architecture, the wire length distribution is discrete. In previous work as well as in this work, the predicted wire length distribution has often been evaluated by comparing it to the normalized histogram of the wire lengths resulting from *one or multiple placement experiments* of a circuit (Figure 2.4). This is based on an implication of the homogeneity assumption, namely that the wire lengths are *ergodic*, i.e., that the length of each individual wire, considered across multiple placement runs, follows the same statistics as the length of a randomly selected wire in a single placement. A weaker form of evaluation consists of comparing the mean of the predicted wire length distribution to the measured average of the wire lengths from one or multiple placement runs.

⁵With respect to real circuits, which do contain multi-terminal nets, what we actually try to predict is the distribution of the source-to-sink lengths.

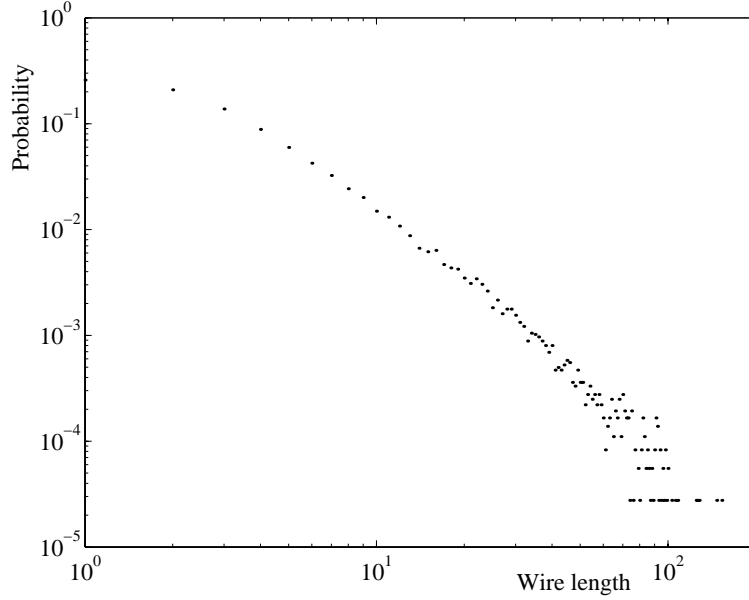


Figure 2.4: Wire length distribution obtained from a well-optimized placement of benchmark `ibm01`.

Originally, the main target of interconnect prediction was to be able to predict the expected value of the *average wire length*, i.e., the expected value of the length of a randomly selected net, resulting from a given randomized placement process \mathcal{P} . However, more recently, the non-linear impact of wire length on several implementation cost metrics, such as interconnect delay or power dissipation, has drastically increased. To obtain more accurate a priori estimates of those metrics, the wire length distribution is required. This wire length distribution, as well as the corresponding value of the homogeneous placement cost C that can be achieved in a particular placement experiment, are the result of the combined effects of

- (i) the heuristic that is used to optimize the cost function C ,
- (ii) the placement architecture and
- (iii) the circuit interconnect topology.

In what follows, we demonstrate the impact of each of these three aspects on the total wire length C .

2.5 Analysis of placement results

2.5.1 Impact of placement heuristics

Throughout the years, many different placement heuristics have emerged. Their common goal is to produce a reasonably good placement in an acceptable amount of time. In general, good is defined with respect to the overall (combined) cost metric. However, in view of our simplified placement model, we will restrict this discussion to placements which minimize the total wire length C .

For a survey of different placement strategies, we refer to [She02, SWY03]. With respect to our work, they can roughly be divided into *flat* and *hierarchical* approaches. In a flat or *homogeneous* placement technique (closest to the homogeneous placement process defined in Section 2.4), all nets are equally important and optimized with equal effort. At all times during placement, the entire solution space is addressed and changes are made incrementally, often by moving some gates or by *swapping* them (i.e., exchanging their positions). In contrast, a hierarchical placement technique first creates a recursive partitioning of the circuit netlist. After each two-way or four-way partitioning step, the resulting modules are assigned to regions in the placement architecture. Within a hierarchy level, these assignments can then be explored to optimize the total wire length. Hence, at each hierarchy level, only a fraction of the total solution space is explored, often resulting in much faster heuristics. In most commercial placement tools, a combination of flat and hierarchical techniques is applied. First, an initial placement is produced based on recursive partitioning. This relatively good starting point is subsequently optimized by different flat and hierarchical heuristics.

In our experiments, we have used three different placement tools. Two of these, `Qplace` and `Plato`, start from a recursive partitioning based placement. `Qplace` (version 5.0.55, incorporated in Cadence Silicon Ensemble DSM version 5.2) is a commercial placement tool which combines different types of optimizations. Because we had very little knowledge of, and even less control over the optimizations performed in this tool, another recursive partitioning based placement tool, `Plato`, was developed in our research group. On average, it results in larger total wire lengths than `Qplace`, but it allows a tight control over the different optimization stages as well as the measurement of different intermediate results. The third placement tool we used,

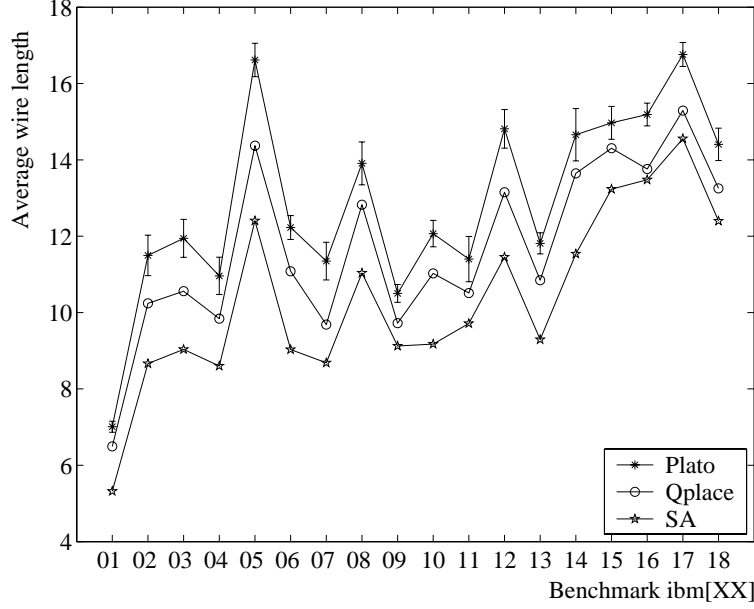


Figure 2.5: Average wire lengths resulting from the three different placement tools used in this Ph.D. thesis (ISPD98 benchmarks). For `Qplace` and `SA`, only a single placement run was performed. For `Plato`, an average was taken across 10 placement runs (3σ error flags are also shown).

performs a flat optimization by *simulated annealing* (SA). The parameters controlling the SA algorithm were chosen such that a very good solution is obtained, but at the cost of an extremely large computation time. Hence, results from this tool also serve as an approximation of the best possible placement that can be obtained for each circuit. These three placement tools are described in more detail in Appendix A.

Figure 2.5 shows the average wire lengths obtained by each of these three tools for benchmarks `ibm01-ibm18`. We can conclude that the placement qualities obtained by different tools can be significantly different. We can expect this differentiation to be even larger if we include the fact that different commercial tools also tend to optimize quite different cost functions. However, as can be seen in Figure 2.5 as well as Table 2.2, the correlation between results from different tools turns out to be quite high. This is particularly the case for different placement tools that use recursive partitioning to construct an initial placement. In this work, we do not go in search for tools that predict the results of a specific tool or tool flow. As a consequence, we will never be able

Table 2.2: Correlations between average wire lengths obtained by different tools, across the 18 ISPD98 benchmark circuits.

	Plato	SA
Qplace	0.9888	0.9715
Plato		0.9538

to obtain an excellent correlation with all three placement tools used in our experiments. In [Ver03], an attempt is made to include the impact of the placement strategy (i.e., hierarchical or flat) on the placement result.

Placement heuristics typically contain some randomized choices that can affect the final placement quality. This effect was called *inherent tool noise* in the “Design” Chapter of the 2001 ITRS [ITR01] and defined as:

“Inherent tool noise is the variation in solution quality that results from non-functional changes to the tool input (e.g. renaming variables or instances, permuting lines of a gate-level netlist, etc.).”

The same source also mentions that noise values of up to 30% have been reported for commercial placement and routing tools but they do not mention which cost metrics this applies to. Clearly, such a high level of tool noise limits the predictability of physical design outcomes and certainly restricts the accuracy that can ever be expected of interconnect prediction techniques.

To quantify the tool noise of Plato, we have performed 10 different placements for each of the benchmarks in Figure 2.5. The error flags in Figure 2.5 represent the 3σ -values resulting from this experiment. In Qplace, there is no direct control over this randomized behavior. However, a quantification based on changing net names was performed in [KM02], and resulted in differences of up to 7% between the best and the worst placement result. For our SA placements, no such exploration was performed because of the already excessive run-times of the tool (up to several months for the largest of the ISPD98 benchmarks).

In all further experiments with Plato, both average wire lengths and wire length distributions were averaged across different placement runs to eliminate the impact of tool noise. A similar averaging is done

when measuring the recursive partitioning Rent characteristics for interconnect prediction.

2.5.2 Impact of the placement architecture

In interconnect prediction, the impact of the placement architecture is captured mainly (but in some cases not only) by *site function(s)* or *site distributions*. In general, given two regions of a placement architecture, \mathcal{G}_1 and \mathcal{G}_2 , the site function $\Phi_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ gives the *number* of pairs (g_1, g_2) of distinct grid positions that are a distance ℓ apart, and for which g_1 belongs to \mathcal{G}_1 and g_2 to \mathcal{G}_2 . Because a grid position pair (g_1, g_2) can contain the two gates a two-terminal net is connected to, it is often referred to as a *(two-terminal) net placement site*, or simply a *site*.

The *site distribution* $\mathcal{D}(\ell)$ is a probability mass function, expressing the probability that a randomly placed two-terminal net has length ℓ . It differs from the site function only by a normalization factor.

A general flow for obtaining site distributions and site functions for rectangular regions is given in Appendix C. In this section, we concentrate on the case where both \mathcal{G}_1 and \mathcal{G}_2 correspond to the entire placement grid. In this case, the site function $\Phi(\ell)$ can be considered as the *capacity* of the architecture for containing (two-terminal) nets of length ℓ . Clearly, when there are more short-distance sites available, a placement tool will probably be able to use more and as a result produce a better placement.

Figure 2.6 and the two graphs in Figure 2.7 show the impact on the site distribution of changing the grid size and the grid and cell aspect ratios, respectively. They show how changing these aspect ratios affect the number of available placement sites that result in sort wires. If this number decreases too much, the placement tool has less freedom for optimization and the final total wire length increases. Figure 2.8 illustrates this effect (example for benchmark `ibm01`, placed with `Plato`). Especially for very long and narrow placement grids, the total placement wire length increases drastically.

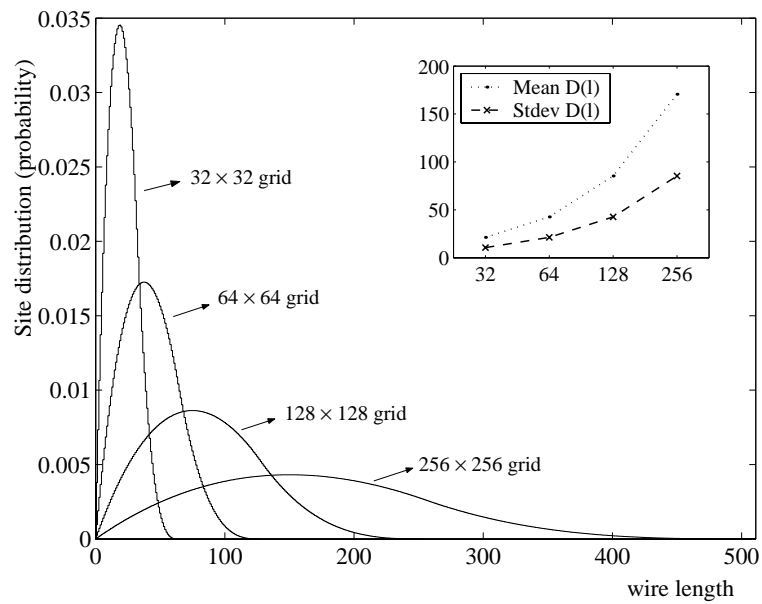


Figure 2.6: Impact of the grid size on the site distribution for square grids with square cells.

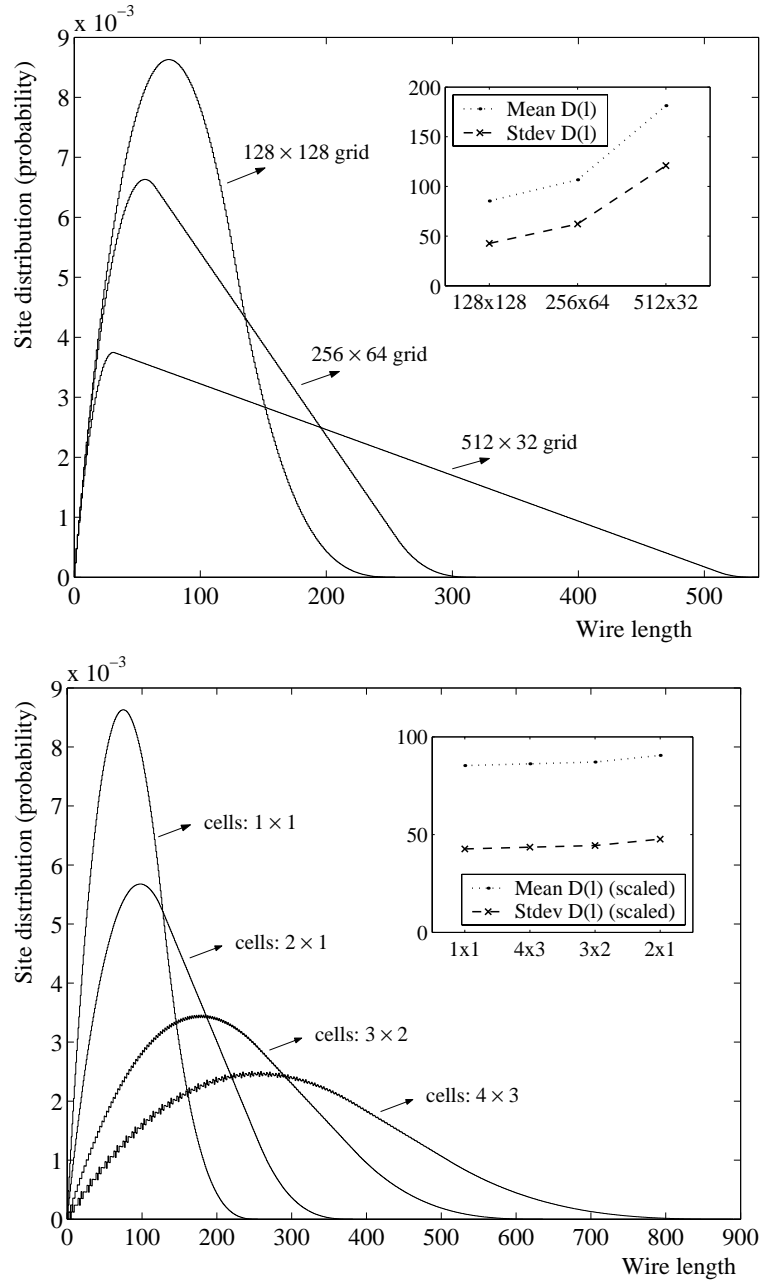


Figure 2.7: Impact of the grid aspect ratio (top) and the cell aspect ratio (bottom) on the site distribution. In the top graph, all grids have 16384 square cells, while in the bottom graph, all grids are square and contain 128×128 cells. For the inlay in the bottom graph, all lengths were renormalized to make cell areas equal. The unit distance used is the square root of this cell area.

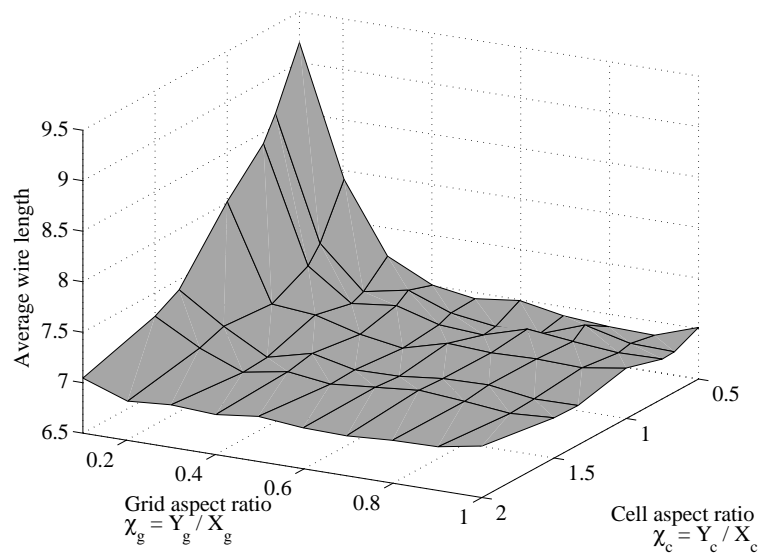


Figure 2.8: Impact of the grid and cell aspect ratio on the average wire length for Plato placements of benchmark *ibm01*.

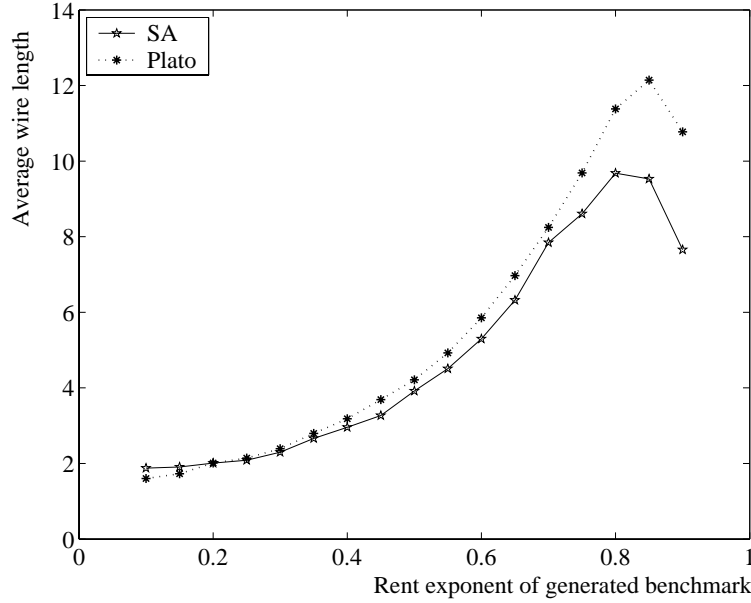


Figure 2.9: Average wire lengths for placed circuits with 16384 gates and controlled values of the Rent exponent p .

2.5.3 Impact of the interconnect topology

Traditionally, in interconnect prediction, the interconnect topology has been characterized by the Rent parameters t and p . Different interpretations exist of how these parameters are to be extracted, but in this section, we restrict our discussion to recursive bipartitioning as it was described in Section 2.2. To illustrate the impact of these parameters, we have generated a series of synthetic benchmarks with `gn1`, which have a partitioning terminal-gate relationship that very closely matches Rent's rule (benchmarks `q10-q90`, see Appendix A). For each of these benchmarks, placements with `Plato` and `SA` were performed. The resulting average wire lengths are shown in Figure 2.9.

This figure illustrates that the Rent exponent p indeed has a very large impact on the average wire length. As we will discuss in Chapter 3, it also affects the general shape of the wire length distribution (in particular, the scaling behavior).

In [Chr00], it was first discussed that a relationship exists between the average number of terminals per gate and the achievable placement quality. In [Ver03], empirical evidence was given to support this. Ad-

Table 2.3: Average block degree, standard deviation of the block degree distribution and measured Rent exponent p for the benchmarks with controlled block degree distribution.

Benchmark	$\text{avg}(t_k)$	$\text{stdev}(t_k)$	p
$n^\circ 1$	5.996	1.875	0.710
$n^\circ 2$	5.779	2.876	0.717
$n^\circ 3$	8.966	3.080	0.712
$n^\circ 4$	8.094	4.706	0.700

ditionally, the standard deviation on the number of terminals per gate was also found to have an impact. However, these effects are smaller than the impact of the circuit's partitioning Rent exponent. As an illustration, we have performed similar experiments. We have used `gn1` to generate a number of benchmark circuits with Rent exponent 0.3, 0.5 and 0.7 and for which all gates have exactly the same block degree. Such benchmarks were generated for $t_G = 4$, $t_G = 8$, $t_G = 16$ and $t_G = 32$. Additionally, we have also generated four benchmarks with $p \simeq 0.7$ and with different block degree distributions. The averages and standard deviations of the number of these block degree distributions, as well as the measured values of p for these benchmarks are shown in Table 2.3. All of these benchmarks were placed with `Plato`, and the resulting average wire lengths were plotted in Figure 2.10. In accordance with the results presented in [Ver03], we can conclude that the average wire length is a slowly increasing function of the average block degree, but that it decreases as a function of the variability of the block degrees in the circuit. Still, the impact of the average block degree and the block degree distributions appears to be smaller than that of the Rent exponent p .

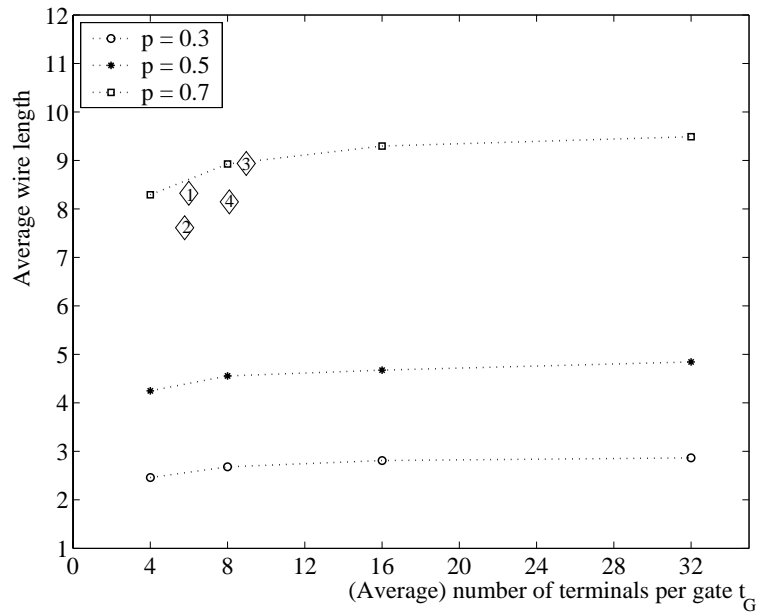


Figure 2.10: Average wire lengths for placed circuits with controlled values of the Rent exponent p , but different values of t_G . Also shown are the average wire lengths for the benchmarks with controlled block degree distribution (large diamonds, with numbers corresponding to the benchmark numbers in Table 2.3).

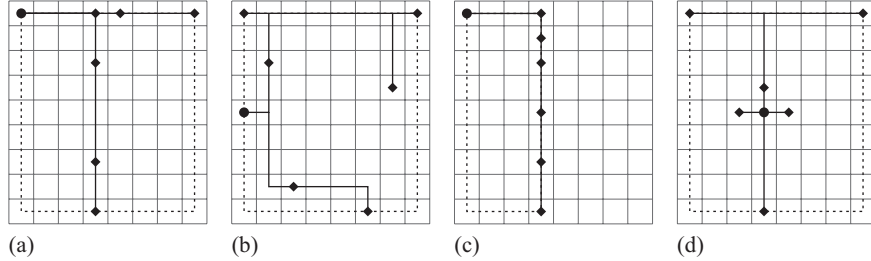


Figure 2.11: Illustration of SMT-routing for several seven-terminal nets. The net source is indicated by a black dot, while the sinks are shown as black diamonds. The dashed lines indicate the nets' bounding boxes.

2.6 Multi-terminal nets

In this section, we discuss multi-terminal nets and their length metrics. We also give some reasons why we think it would be premature in this stage to include multi-terminal nets in our models.

2.6.1 Length metrics for multi-terminal nets

For multi-terminal nets, the post-placement length is given by the length of the *Steiner minimal routing tree* (SMT), i.e., the Manhattan length of the shortest routing tree that connects the source and all sinks (Figure 2.11). Unfortunately, there exists no simple analytical expression to calculate the SMT-length. In particular for large net degrees, the calculation of the SMT can become very time-consuming [She02]. To accord better with the discrete nature of our placement architecture, we will also discretize the possible paths that can be followed for routing. Hence our routing is restricted to a grid of horizontal and vertical tracks that run through the centers of the gate locations in the architecture. Although this restriction greatly simplifies the calculation of SMT's, it still remains too slow for many applications. Because of this, the SMT length is often approximated by other length metrics. The accuracy of these approximations largely depends on the computation time that is available. Although many approximations exist [She02] we will only discuss three that are guaranteed to be either a lower or an upper boundary for the real SMT length.

A lower boundary for the SMT-length of multi-terminal nets is given by half the perimeter of the net's *minimum bounding box* (also

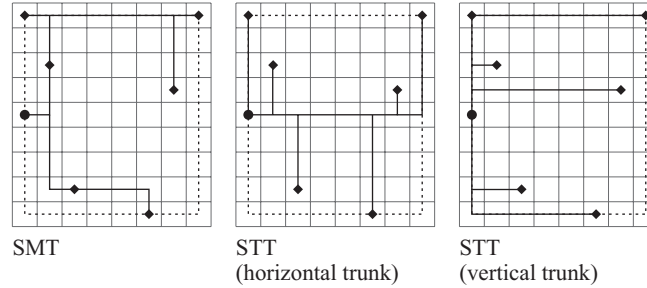


Figure 2.12: Illustration of STT-routing with horizontal and vertical trunks through the net source for net (b) from Figure 2.11.

called the *half perimeter bounding box* or HPBB), i.e. the smallest box in the placement grid that contains all blocks connected to the net. For two- and three-terminal nets, the HPBB is equal to the SMT, while for larger numbers of terminals, the HPBB results in a more severe under-estimation as the net degree increases.

The SMT-length of a multi-terminal net is bounded from above by the sum of all individual source-to-sink distances for all sinks in the net (the *source-sink-length* or SSL). Although the SSL is equal to the SMT-length for two-terminal nets, it is, on average, a very large over-estimation for $k > 2$.

A tighter upper boundary can be derived from the *single trunk* (Steiner) tree or STT [Sou81, CQZC02], which is used as a fast estimator of the SMT. In a STT, the entire maximum routing distance in one dimension is covered by a single *trunk*. When, e.g., a horizontal trunk is chosen, it is placed at the y -coordinate corresponding to the center of gravity of all y -coordinates of the net terminal positions. Obviously, with the grid routing considered here, the routing track closest to the center of gravity must be used. Then, vertical segments are added to connect all terminals to the trunk. Usually, both x -trunk and y -trunk trees are calculated and the best of both is selected. A faster, but less accurate upper boundary can be obtained by considering STT's for which the trunk runs through the net source (Figure 2.12). We will consider only this restricted kind of STT's because, as we will illustrate shortly, their site distribution can be predicted more easily (under certain conditions).

It must be pointed out that estimations of the SMT-length that are based on any one of these metrics can be very inaccurate. Indeed, even

Table 2.4: Length metrics for the four 7-terminal nets shown in Figure 2.11.

	SMT	HPBB	STT horizontal trunk	STT vertical trunk	SSL
(a)	15	15	15	24	39
(b)	23	15	25	29	39
(c)	11	11	11	26	39
(d)	17	15	20	17	22

for a constant SSL or HPBB and a fixed number of terminals, the length of SMT's for different nets can differ significantly, as is illustrated for the examples from Figure 2.11 in Table 2.4.

2.6.2 Placement of multi-terminal nets

For circuits with two-terminal nets only, the interpretation of the total wire length C is very straightforward. However, since the calculation of SMT-length for multi-terminal nets is often considered too slow, many placement tools optimize HPBB, SSL, STT or some other metric instead.

When SSL is optimized, the placement tool will try to place *all* sinks as close to the source as possible. With respect to the total wire length, the optimization of total SSL is equivalent to the optimized placement of a transformed circuit, for which all k -terminal nets have been replaced by $k - 1$ two-terminal nets with the same source. In some cases, this can be a good model for a placement where timing optimization is very important. For instance, in FPGAs, the source-to-sink delay is dominated by the distance between source and sink, while the positions of other sinks have a much smaller impact. This is also the case for longer nets when buffers are inserted.

The optimization of HPBB mainly results in small routing areas for the entire net and, as a consequence, smaller equivalent net capacitances. However, in this case, only the edges of the bounding box have an impact on the cost function, while all other terminals can be placed randomly within the bounding box region.

The fact that many different length metrics as well as derived cost functions are used in placement tools, is an additional argument to doubt the usefulness of a very accurate modeling of multi-terminal nets. Hence, for the remainder of this chapter, we will only consider

two-terminal nets. Since all our experiments were performed on circuits which originally did contain multi-terminal nets, but for which they were replaced by two-terminal nets as described above, our cost metric effectively corresponds to total SSL for both the transformed *and* the original circuits.

Finally, there exists no real consensus on the question whether SMT-length is even an accurate estimation for actual routing length. If this is not the case, there is no reason to even attempt a good estimation of SMT-lengths. To our knowledge, a systematic exploration of the correlation between post-placement length metrics and routing length (as a function of net degree) has not yet been published.

Chapter 3

Fundamental approaches to interconnect prediction

In this Ph.D. thesis, we want to predict wire length distributions that can be used to predict and compare the quality of different design options during digital design. This requires a good correlation of the derived metrics across both topological and architectural variants. In this chapter, we study the two most frequently used types of interconnect prediction techniques: those based on recursive partitioning (Donath's interconnect prediction technique, Section 3.2) and those based on Rent parameters extracted from a circuit layout (Section 3.3). The main representative of this last category is Davis' concept of terminal conservation, which is discussed in Section 3.3.2. Both approaches are based on different interpretations of Rent's rule. For each approach, we explain the basic principles, as well as recent extensions by other authors. The state-of-the-art prior to this Ph.D. research is evaluated based on its ability to predict wire length distributions for different placement architectures and circuit topologies, using only parameters that can be determined prior to placement. The quality of those predictions will be evaluated by the simplest derived metric, average wire length, and correlations between predicted and measured values of this metric. Then, our findings are combined into a comparative discussion of both categories of prediction techniques.

3.1 Introduction

The basis for most traditional interconnect prediction techniques is the assumption that the terminal-gate relationship of circuit modules can

be approximated by Rent's rule. In every application of this rule, it was understood that the modules under consideration were not random, but that in some sense their number of terminals was minimized. However, the constraints regarding which modules were allowable were not always very strict.

In 1999 (at the first SLIP workshop [SLI99]) it became clear that, over the years, two fundamentally different approaches to module generation had evolved [CS00]. On the one hand, Donath's prediction technique and methods derived from it focus on modules that are generated by directly partitioning the circuit graph according to a pin minimization criterion. Because they are derived only from the graph topology, we will refer to them as *topological* circuit modules.

On the other hand, several authors concentrated on the geometric aspect of module generation [DDM98, RFCR99] (i.e., *geometric* modules), using layout partitioning or generally regions cut from a circuit layout to generate the data points of a Rent plot. The concept of applying Rent's rule to regions from a circuit layout was first introduced by Feuer [Feu82]. He argues that a region from a circuit layout creates, in a sense, a partitioning of the circuit, separating the gates within the region from those outside it. In a good placement the wires are kept as short as possible. Hence, the wires crossing any boundary in the resulting layout will originate mainly from gates close to that boundary. Since this is only a small fraction of all possible gates, the number of wires crossing the boundary is effectively minimized. This also applies to the closed boundary of a layout region: the number of wires that connect gates within the region with gates outside it (or the outside world), i.e. the number of module terminals, should be small. From this argument, he decides that the partitioning created by a layout region from a good placement, is a good partitioning.¹ Since he assumes that all good partitionings result in modules with a terminal-gate relationship that obeys Rent's rule, this must also be the case for geometric modules.

The topological and the geometric interpretation of Rent's rule have resulted in two different classes of interconnect prediction techniques. Donath introduced his hierarchical partitioning based (and hence topological) model for wire length estimation in 1979 [Don79]. Since then, it has become one of the most frequently used models for a priori wire length prediction.

¹A good partitioning generally refers to a partitioning for which the number of nets cut is minimized.

The most frequently used geometric wire length prediction technique, is the one first presented in [DDM98] by Davis et al. In this chapter, we will review the original versions of those techniques, identify their main causes of inaccuracy and evaluate their usefulness for the prediction of individual placement results.

3.2 Donath's approach to estimating wire length distributions

As we discussed before, some of the earliest tools for automated circuit placement were based on recursive circuit partitioning. Essentially, Donath proposed a stochastic model for a stripped version of this placement approach.

In this section, we first discuss Donath's original model. Since this model was based on several rather restrictive assumptions, we first validate it in an experimental setup where all these assumptions are met. Subsequently, we also validate it under more realistic conditions. Finally, we briefly discuss several extensions to Donath's model that try to relax some of its restrictions.

3.2.1 The original model

In Donath's original paper [Don79], the aim was to predict the average wire lengths in computer systems, as a function of the circuit's partitioning properties, modeled by Rent's rule. Essentially, he proposed a model for a hierarchical, 4-way partitioning based placement approach, where optimization is achieved only during the partitioning stage. His architectural model is a square grid of 4^H gate positions that serves as a layout medium for a circuit with $G_c = 4^H$ gates.

In a first step, both the circuit and the architecture are partitioned into four equal parts of 4^{H-1} gates and gate positions, respectively. For the architecture, this is done by two symmetrical orthogonal cuts (one in the x-direction and one in the y-direction). Then, each circuit submodule is assigned to a section of the architecture as shown in Figure 3.1. Since all four circuit submodules, as well as all four architecture sections are assumed to be equal, this assignment is essentially random.

This procedure is repeated in a hierarchical way: at each hierarchy level k ($k = H$ down to 1), all 4^{H-k} modules of 4^k gates are partitioned,

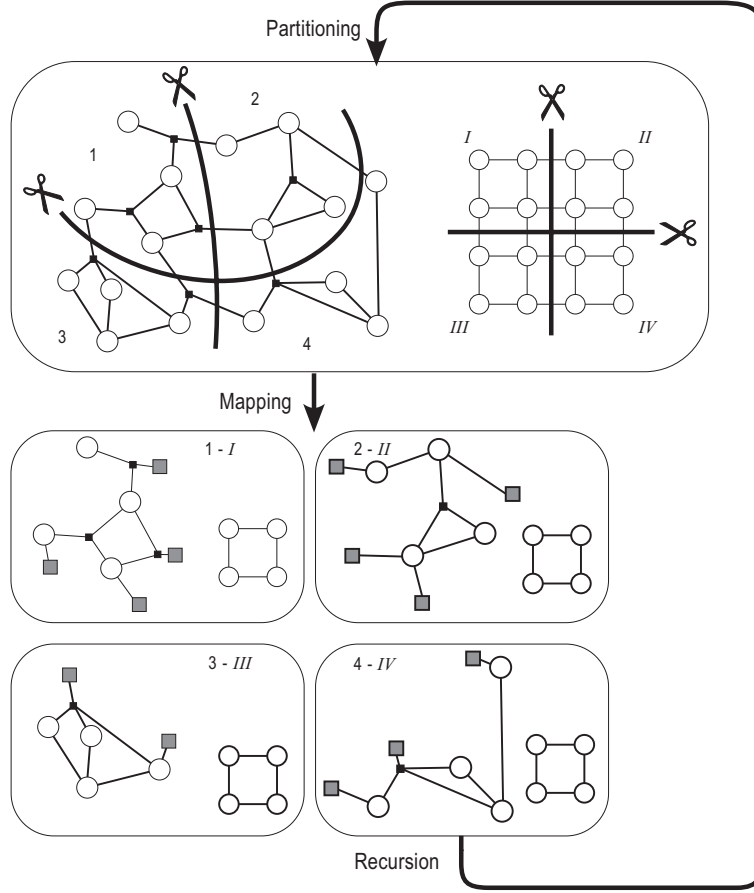


Figure 3.1: Illustration of Donath's placement model, based on hierarchically partitioning circuit and architecture and mapping the circuit modules to the architecture modules.

until finally, at level $k = 0$ all circuit modules consist of a single gate only.

For each module, the number of external terminals is derived from Rent's rule. Therefore, using the concept of terminal conservation [DDM98], the number of nets $N_{cut,k}$ that is cut when partitioning a module of $G_k = 4^k$ gates can be calculated:

$$\begin{aligned}
 N_{cut,k} &= \frac{4T(G_{k-1}) - T(G_k)}{2} \\
 &= t \frac{4G_{k-1}^p - G_k^p}{2}
 \end{aligned} \tag{3.1}$$

3.2 Donath's approach to estimating wire length distributions 53

$$= t4^{(k-1)p}(2 - 2^{2p-1}). \quad (3.2)$$

To determine the length distribution of those connections, Donath assumed their terminals to be homogeneously spread across the respective modules. This corresponds to a random assignment of the modules, or a uniformly constant *occupation probability* $Q(\ell)$ of the sites within each hierarchy level. Thus, the length distribution of the nets that are cut at level k is a fraction of the *site function* at that level, i.e., the distribution of the distances between gate pairs for which one gate is in one submodule and the other is in the second submodule. Because of the way the architecture is partitioned, two different site functions are required: one for neighboring submodules and one for diagonally opposed submodules. For detailed calculations of these distributions, we refer to [Str01b]. We will only mention their averages at level k , $\mu_{k,n}(\ell)$ and $\mu_{k,d}(\ell)$, for neighboring and diagonally opposed submodules, respectively:

$$\begin{aligned} \mu_{k,n}(\ell) &= \frac{4}{3}\lambda - \frac{1}{3\lambda} \\ \mu_{k,d}(\ell) &= 2\lambda \end{aligned} \quad (3.3)$$

where λ equals 2^{k-1} , the side of the submodules that result from partitioning the architecture module at level k .

For 4-way partitioning, two pairs of submodules are diagonally opposed and 4 pairs are neighboring. The average wire length at level k is therefore given by:

$$\begin{aligned} \mu_k(\ell) &= \frac{4\mu_{k,n}(\ell) + 2\mu_{k,d}(\ell)}{6} \\ &= \frac{14}{9}\lambda - \frac{2}{9\lambda} \end{aligned} \quad (3.4)$$

Finally, given the number of modules 4^{H-k} that is partitioned at each level, the global wire length distribution can be calculated, as well as an analytical expression for its average [Str01b]:

$$\begin{aligned} \mu(\ell) &= \frac{\sum_{k=1}^H 4^{H-k} N_{cut,k} \mu_k(\ell)}{\sum_{k=1}^H 4^{H-k} N_{cut,k}} \\ &= \frac{14}{9} \frac{F(H, p, 1)}{F(H, p, 2)} - \frac{2}{9} \frac{F(H, p, 3)}{F(H, p, 2)} \end{aligned} \quad (3.5)$$

where $H = \log_4(G_c)$ is the number of hierarchy levels, p is the Rent exponent, and $F(H, p, x)$ is given by:

$$F(H, p, x) = \frac{2^{H(2p-x)} - 1}{2^{2p-x} - 1} \quad (3.6)$$

Donath's model can just as easily be applied to predict the entire wire length distribution. Indeed, when the actual site distributions are used instead of their averages, Donath's model results in a prediction of the wire length distribution:

$$\mathcal{N}(\ell) = \sum_{k=1}^H \frac{4^{H-k} N_{cut,k}}{N_c} \mathcal{D}_k(\ell) \quad (3.7)$$

where $\mathcal{D}_k(\ell)$ represents the site distribution at level k , composed of the site distributions for neighboring module pairs and diagonally opposed module pairs, $\mathcal{D}_{k,n}$ and $\mathcal{D}_{k,d}(\ell)$, respectively, as follows:

$$\mathcal{D}_k(\ell) = \left(\frac{4}{6}\right) \mathcal{D}_{k,n}(\ell) + \left(\frac{2}{6}\right) \mathcal{D}_{k,d}(\ell). \quad (3.8)$$

This is illustrated in Figure 3.2 for $k = 7$.

In [Don81], Donath found that the resulting distribution has a power law scaling behavior $\mathcal{N}(\ell) \sim \ell^{2p-3}$ for a wide range of lengths ℓ , with deviations occurring for the shortest as well as the longest lengths. This was later confirmed by Cotter et al. in [CC91]. These authors also derived an analytical expression for the site distributions $\mathcal{D}_k(\ell)$ and verified it for two industrial designs.² In his PhD thesis, Stroobandt developed a very elegant argument to support this scaling behavior. In fact, his argument was not restricted to regular, 2-dimensional grids. Instead, he assumed a D -dimensional grid with the number of grid positions in each dimension equal to 2^H . He found the generalized scaling behavior to be:

$$\mathcal{N}(\ell) \sim \ell^{Dp-(D+1)}. \quad (3.9)$$

We should mention here that in Stroobandt's derivation, Equation (3.9) was obtained by making no assumptions regarding the specific form of the local wire length distributions at each hierarchy level, other than that their average scales as 2^k .

²Note that, in [CC91], no mention was made of how the Rent exponent p was estimated.

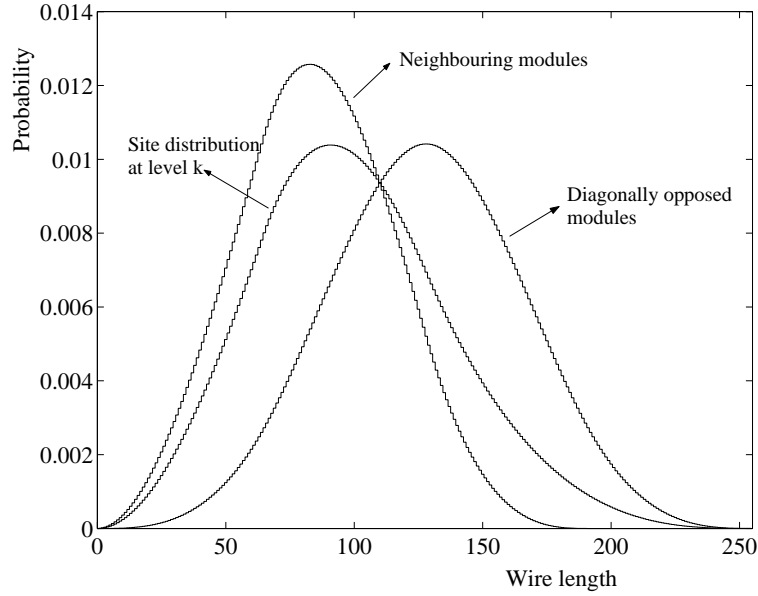


Figure 3.2: Illustration of the composition of the site distribution $\mathcal{D}_k(\ell)$ at level k , for $k = 7$. As can be seen, the site distribution for neighboring module pairs has a smaller average than that for diagonally opposed module pairs.

3.2.2 Validation of the original model

In this section, we validate Donath's model in several ways. First, we ascertain that it is accurate when all of its underlying assumptions are met. Under these conditions, we also verify the power law scaling behavior of the wire length distribution and compare it to the scaling laws found in strongly optimized placements. Then, we apply Donath's model to industrial circuit placements.

The boundary conditions for Donath's model are the following:

1. *the number of gates in the circuit is a power of four and equals the number of grid positions in the architecture;*
2. *the architecture grid is square and has equal unit distances in both directions .*
3. *placement is based on hierarchical 4-way partitioning;*
4. *placement optimization is done only by optimized partitioning, i.e. the four circuit modules are randomly assigned to the four architecture modules at each partitioning stage;*

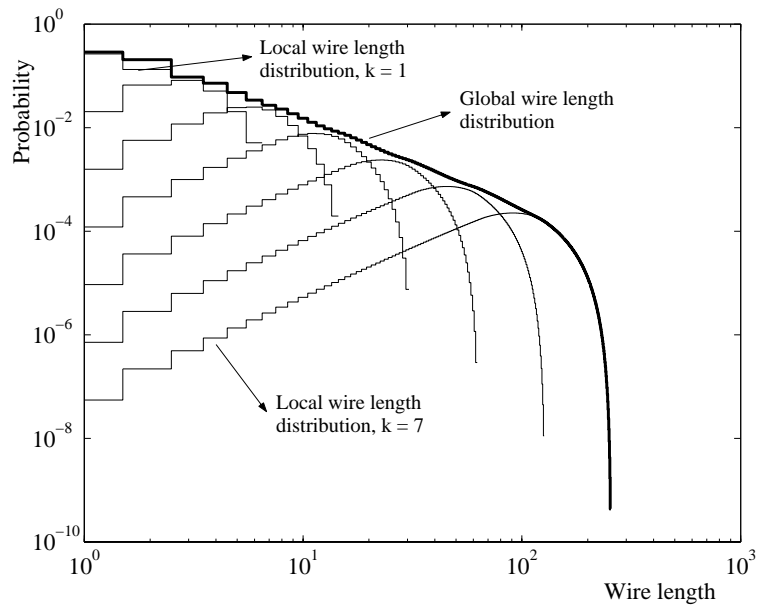


Figure 3.3: Illustration of the combination of local site distributions into the global wire length distribution as given by Equation(3.7), with $H = 7$ and $p = 0.65$.

5. *the circuit partitioning properties can be described by Rent's rule for all module sizes ;*
6. *circuits have two-terminal nets only;*
7. *only internal nets are modeled;*

Note that the last two boundary conditions are in accordance with the models and assumptions adopted in this work (Chapter 2). This is definitely not the case for conditions 1 and 2, which are more restrictive than our own architecture model. Of condition 5, we know that it is only a rough approximation of reality.

To meet Donath's boundary conditions, we have used the synthetic benchmarks `q10-q90` (see Appendix A). These benchmarks have 16384 gates each and $t_G = 4$. They were generated by `gnl` in such a way that their partitioning terminal-gate relationship follows Rent's rule as closely as possible, with values of p ranging from 0.1 to 0.9 (increments of 0.05). They were placed in a square grid architecture, with $X_g = Y_g = 128$ ($H = 7$) and $X_c = Y_c = 1$, by recursive 4-way partitioning based placement. Since `gnl` generates benchmarks in a *bottom-up* way by hierarchically clustering modules, a clustering tree can be saved during the generation process. The same tree was then used as a partitioning tree for placement. Because we forced the module sizes at every level to be equal, the resulting partitioning tree is perfectly balanced.

On one point we could not exactly mimic Donath's model: `gnl` generates circuits by recursive bi-clustering. This means that the modules in our partitioning tree do not correspond to a recursive quadrisecion, but to a recursive bisection. However, it can be verified that a bisectioning, followed by a random assignment of the modules to any of the four positions (I, II, III, and IV in figure 3.1), will, on average, result in the same average wire length as a quadrisecion.

To obtain this stochastic average, 500 placement runs were performed for each circuit. From the resulting average wire lengths, the average and standard deviation were calculated. These results, together with the average wire length estimation resulting from Donath's model, are shown in Table 3.1. Figure 3.4 shows the predicted and average wire length distributions for two of these benchmarks.

Both Table 3.1 and Figure 3.4 show that indeed, if all of Donath's assumptions are met and if statistical variation is eliminated, Donath's model provides an accurate estimate of the average wire length. Still, there remain some differences between the model and the experimental

p	Prediction (Donath)	Experiment (Plato-Quad)	
	$\mu(\ell)$	$E[\ell]$	$stdev[\ell]$
0.10	2.37	2.33	0.011
0.15	2.53	2.45	0.014
0.20	2.72	2.77	0.018
0.25	2.96	3.00	0.020
0.30	3.25	3.29	0.025
0.35	3.61	3.56	0.033
0.40	4.07	3.97	0.044
0.45	4.63	4.68	0.058
0.50	5.34	5.33	0.075
0.55	6.22	6.15	0.100
0.60	7.32	7.43	0.134
0.65	8.67	8.69	0.171
0.70	10.33	10.28	0.223
0.75	12.33	12.39	0.286
0.80	14.72	14.67	0.359
0.85	17.51	17.38	0.449
0.90	20.70	20.62	0.536

Table 3.1: Average wire length results for synthetic benchmarks that follow Rent's rule almost exactly: predicted results for the average wire length $\mu(\ell)$ from Donath's model (3.5) and experimental values of average and standard deviation of $E[\mathcal{N}(\ell)]$ across 500 placement runs.

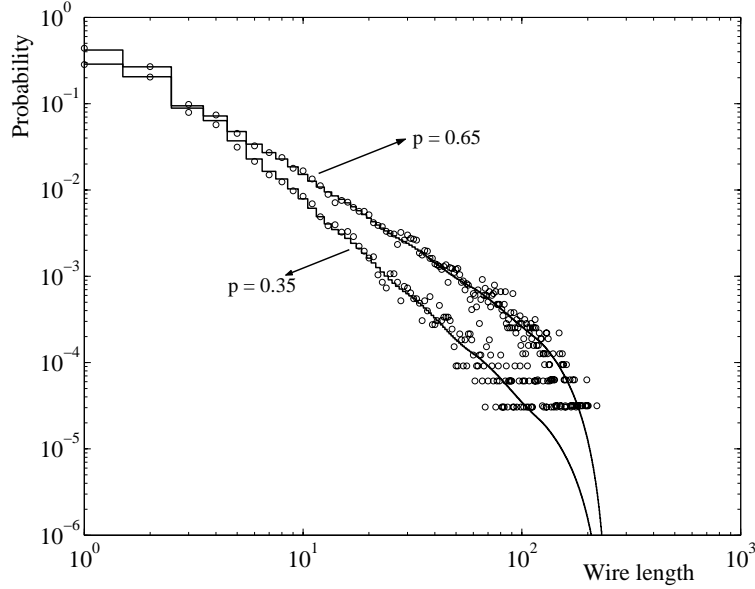


Figure 3.4: Predicted wire length distribution from Donath's model for $p = 0.35$ and $p = 0.65$, as well as the results from a single Plato-Quad placement run of benchmarks q_{35} and q_{65} , respectively.

results. These can only be due to the single difference between theory (Donath's model) and experiment, namely the fact that the benchmark partitioning Rent characteristics are very close to Rent's rule, but do not exactly match it.

Hence, we have ascertained that there exists a (partially) optimized placement process and a set of benchmarks for which Donath's model is almost perfect. But what is the relation between the results of this placement process and the results of other placement processes that perform a more extensive optimization? To answer this question, we have also performed placements of the benchmarks q_{10} - q_{90} with Plato (average of 10 randomized placement runs) and with SA (simulated annealing, one placement run only). The results of these experiments are shown in Figure 3.5.

Clearly, these placement approaches yield better placement results (at the cost of increased computation time). However, as long as the Rent exponents are not too high (below 0.8), the resulting average wire lengths correlate well with the predicted values and the results from the Plato-Quad placement. Only for very high Rent exponents, de-

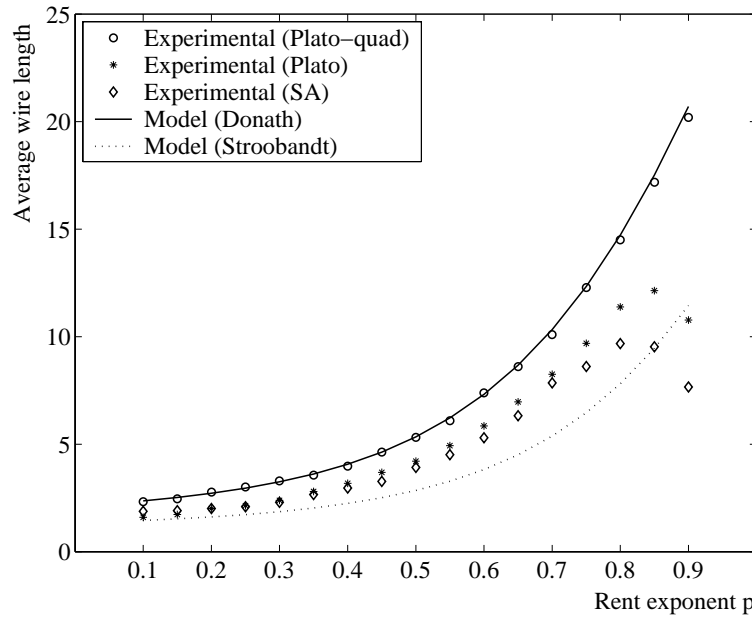


Figure 3.5: Average wire lengths resulting from the three placement approaches Plato-Quad, Plato and SA for benchmarks q_{10} - q_{90} , as well as the average wire lengths resulting from Donath's model and from Stroobandt's model for optimized placement, using the occupation probabilities from Equation (3.13).

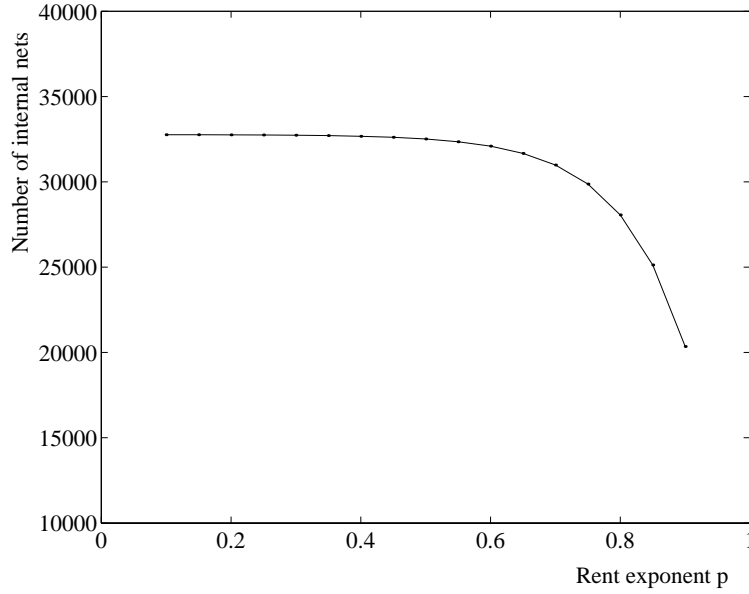


Figure 3.6: Decrease of the number of internal nets for benchmarks $q10$ - $q90$.

viations occur. We expect that this is due to the fact that for such high Rent exponents, the number of external nets becomes very high. Therefore, since the total number of gate terminals is the same for all of these benchmarks, only very few internal nets remain (see Figure 3.6). Because the external nets are not included in Donath's model, their lengths were also not optimized during placement. We can expect this situation—the optimization of only very few nets—to be exceptionally easy for a placement engine, which explains why the average wire lengths are much shorter than expected. Remember that, as we discussed in Section 2.5.3, Verplaetse [Ver03] experimentally verified that, for strictly Rentian behavior, the number of terminals per gate does affect the average wire length. Hence, we expect similar experiments with a constant number of internal nets per circuit (i.e., a constant number of internal terminals per gate) would not show this artifact. However, we did not have time to perform these experiments as well.

To take our evaluation of Donath's model one step further, we have also performed experiments on the more realistic *ispd98* benchmark circuits *ibm01* - *ibm18*. Strictly speaking, Donath's model is not ap-

plicable to those circuits, because the model conditions can usually not be fulfilled. Amongst other things, for none of them does G_c equal a power of four. To come as close to Donath's model as possible, we have therefore performed a placement in a square grid with $H = \lceil \log_4(G_c) \rceil$. As in the previous experiments, the circuit modules were randomly assigned to the architecture modules at each level. For each benchmark, an average was taken across 100 placement runs.

Then, the grid was slowly compacted to its final size of $X_g \times X_g$, corresponding to an approximate occupation of 95% of the grid locations. This compaction was performed in such a way that the relative positions (the two-dimensional *ordering*) of the blocks were disturbed as little as possible (see Appendix A). If this compaction were perfect, it would result in a downscaling of all wire lengths by a factor

$$\eta = \frac{X_g}{2^H}. \quad (3.10)$$

Obviously, such a compaction is generally not possible, since, as the fraction of occupied grid locations increases, more and more contention will occur. Still, as is illustrated in Figure 3.7, our compaction algorithm performs very well, in the sense that the final average wire length is only slightly above the expected (scaled) average wire length.

To model the initial placement stage, predictions were obtained from Equation 3.4, using the Rent exponents shown in Table 2.1 and the same value of H as was used for placement. Then, the scaling factor η was applied to the resulting average wire length. Figure 3.7 shows both the predicted and the experimental results, with and without compaction, as well as strongly optimized experimental results obtained with the three different placement tools: `Plato`, `Qplace` and `SA`.

The figure shows that, although there seems to be some correlation between the predicted average wire lengths and the placement results they are trying to mimic for some of the benchmarks, the predicted values are an enormous under-estimation. In fact, they are much closer to the results obtained from placement tools that perform a high level of optimization.

From the first series of experiments, we found that Donath's technique can predict the placement results of one particular placement approach for strictly Rentian benchmark circuits. However, this is no longer the case for "real" circuits, even when Donath's placement approach is mimicked as closely as possible. For these benchmarks, we find that the predicted results strongly underestimate the experimen-

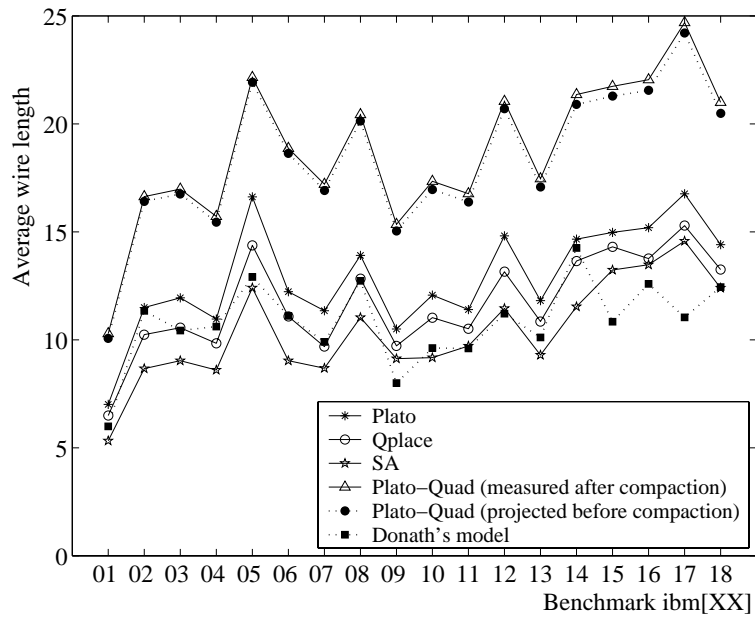


Figure 3.7: Average wire lengths for the ISPD98 benchmarks, resulting from a placement that is as close to Donath's model as possible, as well as the predictions from Donath's model. For comparison, the results obtained by Qplace, Plato and SA are also shown.

tal results they were meant to mimic. Even worse is the fact that the predictions do not seem to correlate well with the experimental results. Although in these experiments, we could not exactly match the experimental placement process with Donath's placement model, the resemblance between them was still very close. A much stronger deviation from Donath's assumptions can be found in the differences between Rent's rule and the partitioning Rent characteristics of the `ispd98` circuits. Hence, we can assume that these deviations are the main cause of the observed inaccuracy in Donath's model, when applied to industrial benchmarks circuits.

3.2.3 Extensions to Donath's model

In this section, we briefly summarize and discuss several extensions to Donath's model that were proposed in literature. Each of these extensions tries to alleviate one or multiple restrictions in Donath's model.

Anisotropic grids

The first three restrictions in Donath's architectural models were partially alleviated by Van Marck in [VM00, VMSVC95a, SVC97] as well as in [MY87]. Van Marck studied three-dimensional grids, with the number of grid positions in each direction still powers of two, but not necessarily equal. He also allowed the cell pitches in each dimension to be different (in our own architecture model, this corresponds to non-square or non-cubic cell dimensions). Instead of using four-way partitioning at each hierarchy level, he allowed the choice between:

- two-way (one-dimensional) partitioning, cutting through the vertical direction,
- four-way (two-dimensional) partitioning, cutting through the X - and Y -dimensions,
- eight-way (three-dimensional) partitioning, cutting through all dimensions

To decide on these partitioning types and their ordering, he tried out different possibilities and chose that which resulted in the smallest average wire length. Obviously, the resulting partitioning schedule depends on the module dimensions and the cell pitches.

3.2 Donath's approach to estimating wire length distributions 65

His extensions were used to study three-dimensional opto-electronic architectures, where the X - and Y -dimensions were provided by traditional electric planes, while the third dimension was provided by point-to-point optical interconnects, leaving directly from the surface of the chip planes, instead of from their perimeter. He used the parameter Z_c to model the different cost of the optical connections, compared to that of nearest-neighbor on-chip connections.

Optimization

The fourth restriction of Donath's model, the assumption regarding the placement process, was tackled by Stroobandt in [Str01b]. He stated that the assumption of having a uniform occupation probability within each hierarchy level does not reflect a well optimized placement. Indeed, one would assume that even within a hierarchy level, there would be much more short inter-module connections than long ones.

To correct this, he derived an expression for the global occupation probability from the scaling behavior (3.9). It was based on the observation that, for short wire lengths, the site distribution $\mathcal{D}(\ell)$ is approximately proportional to ℓ . Because in optimized placements, the wire length distribution is dominated by short wire lengths, Stroobandt targeted a global occupation probability that was most accurate for small values of ℓ . For these lengths, he found that the global site function $\Phi(\ell)$ of a square, two-dimensional placement grid is approximately linear in ℓ . In combination with the global scaling behavior of $\mathcal{N}(\ell) \sim \ell^{2p-3}$, the scaling behavior of the global occupation probability (for two-dimensional placement grids) is found to be:

$$\mathcal{Q}(\ell) = \frac{N_c \mathcal{N}(\ell)}{\Phi(\ell)} \quad (3.11)$$

$$\sim \frac{\ell^{2p-3}}{\ell} = \ell^{2p-4} \quad (3.12)$$

Remember that $\mathcal{N}(\ell)$ represents a probability mass function, which is obviously normalized. In contrast, $\mathcal{Q}(\ell)$ determines a series of probabilities which is *not* normalized.

His next step is to assume that the probability that a net placement site is actually occupied by a net depends only on the length of that net and not on the specific location of the site. Hence, the occupation probability within each hierarchy level should follow the same scaling

law as the global occupation probability:

$$\mathcal{Q}_k(\ell) \sim \ell^{2p-4} \quad (3.13)$$

and therefore:

$$\mathcal{N}_k(\ell) = C_k \mathcal{Q}(\ell) \Phi_k(\ell). \quad (3.14)$$

The constants C_k are required to guarantee that the local wire length distributions are normalized.

Stroobandt subsequently uses a continuous approximation to obtain analytical expressions for the averages of the local wire length distributions $\mathcal{N}_k(\ell)$ [Str01b] to replace the Expressions (3.4) in Equation (3.5). The total average wire lengths, resulting from his model, are included in Figure 3.5. Figure 3.8 compares the resulting wire length distributions to those from Donath's original model and the distributions found from a very well optimized placement (SA) for benchmarks `q35` and `q65`.

Multi-terminal nets

The impact of multi-terminal nets on the total wire length and the wire length distribution has always been a topic of discussion. It is obvious that they are important, since (as we discussed in Section 2.6.2) they allow a drastic reduction of routed wire length for the same number of source-to-sink connections. However, the modeling of their impact is very difficult.

In [Str01b], Stroobandt proposed a model for the recursive partitioning properties of multi-terminal nets. This model was integrated in his prediction approach (i.e., Donath's model with the occupation probability extension) to predict length distributions for all net degrees. Because this model is mathematically rather extensive, we will not discuss the details here, but refer to [Str01b].

External nets

External nets were not considered in Donath's original placement model. However, an extension to include them is not very difficult. Several solutions are possible, but we will restrict this discussion to that which remains closest to Donath's framework. In [VM00, SVMVC97],

3.2 Donath's approach to estimating wire length distributions 67

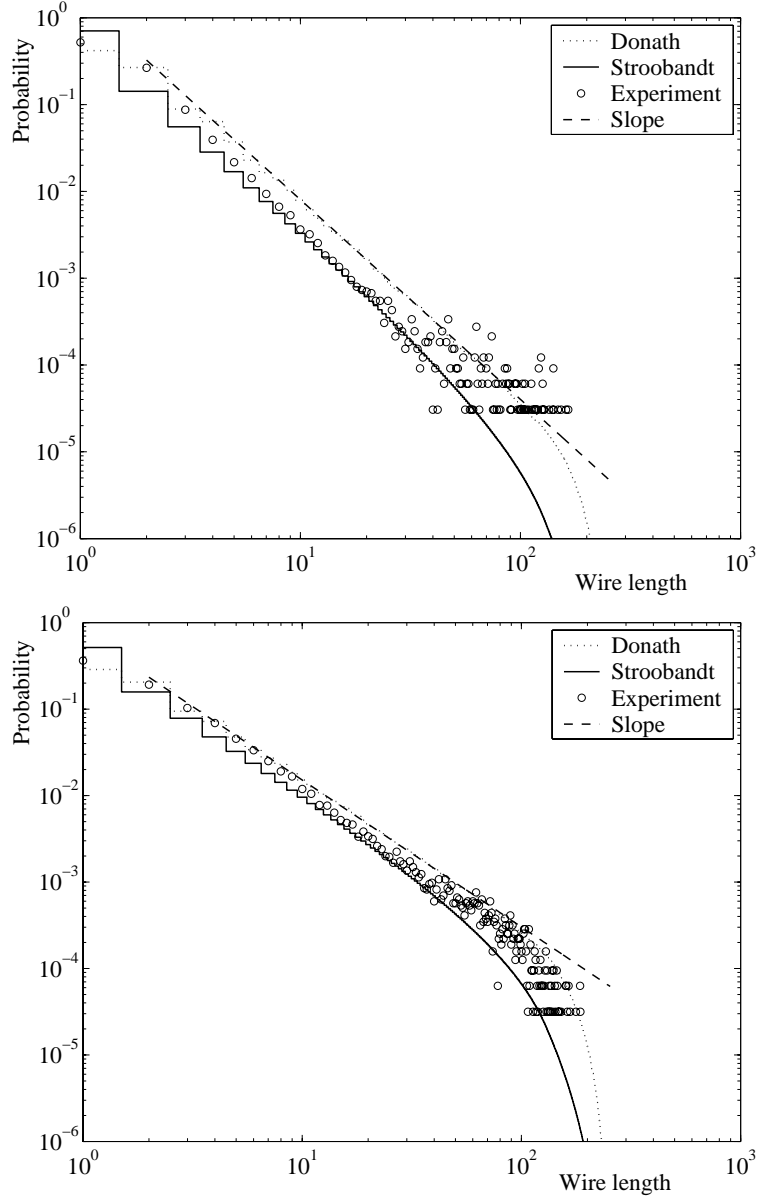


Figure 3.8: The global wire length distribution obtained with Stroobandt's occupation probability (Equation 3.13) for benchmark q_{35} (top) and q_{65} (bottom). Also shown are the wire length distributions resulting from Donath's original model and their power-law scaling behavior $\mathcal{N}(\ell) \sim \ell^{2p-3}$, as well as the experimental wire length distributions obtained by SA.

the I/O-pads are assumed to be placed in a ring around the chip placement grid. Hence, external connections are wires between the placement grid and the I/O-ring. Their number is dictated by Rent's rule, i.e.:

$$T_{ext} = tG_c^p \quad (3.15)$$

If we remain strictly within Donath's framework, we can perform one level of quadrissection, with a quarter of the external terminals assigned to each of the four quadrants. Then, for each quadrant, we should assume both the starting points and the end-points to be homogeneously distributed in their respective regions.

Note that, as we have pointed out in Section 2.2, Equation (3.15) yields a gross over-estimation of the number of external connections for high values of the Rent exponent. In reality, in most designs, the number of external connections is only a small fraction of the total number of connections. Therefore, we might hope that the inclusion of their length distribution in the total wire length distribution will not have a very large impact. Figure 3.9 illustrates this for benchmark circuit s38584 from the LGSynth benchmark series, which was placed with Plato with the external net lengths included in the overall cost function. The external nets clearly have a larger average length (17.5, compared to 7.79 for the internal nets), but their number is relatively small and the global wire length distribution is very close to the length distribution of the internal nets. Still, they do have an impact on the average wire length (in this case, an increase of approximately 5%). Hence, although we will not do it in this work, it would be best to include external nets in future interconnect prediction models, especially when modeling, e.g., IP-blocks with a large number of I/O-signals.

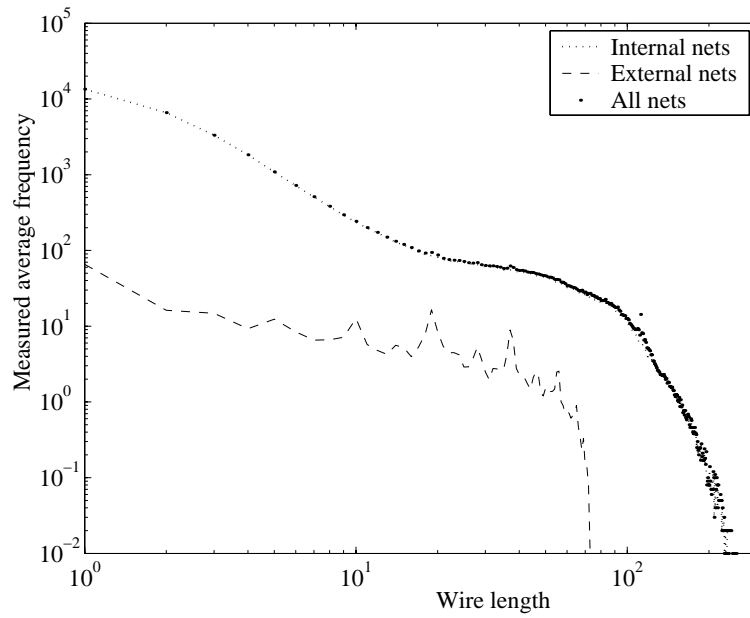


Figure 3.9: Wire length distribution for the internal nets (avg.= 7.79), the external nets (avg.=17.5) and all nets (avg.=8.17), resulting from a `Plato` placement of benchmark `s38584`.

3.3 Geometric techniques for interconnect prediction

The basis of geometric prediction techniques is the application of Rent's rule to modules that are created by "cutting" a region from an optimized circuit layout. All gates within the region boundaries belong to the module, while all gates outside it do not.

Starting from the assumption that Rent's rule does apply to geometric modules, geometric prediction techniques derive what the wire length distribution must look like to fulfill this assumption. In a sense, this is a sort of reverse reasoning, starting from an assumption about the result of a good placement, instead of trying to model an actual placement process (as Donath did). Because the relationship between the circuit topology and the terminal-gate relationship for geometric modules is not clear, the Rent exponent in geometric prediction models should be interpreted as a model parameter. Often, it is determined by fitting the prediction models to experimental results for specific classes of circuits (e.g., from particular application domains). The value of p so obtained is then considered to be typical for that class.

In this section, we discuss the most important representatives of this category. First, we briefly address Sastry's approach [SP86]. Then, we continue to Davis' approach [DDM98], which has been the basis for a very large fraction of the Rentian interconnect prediction techniques we know today.

3.3.1 The first geometric wire length distribution models

The first to apply Rent's rule to layout regions was Feuer [Feu82]. He used a continuous architecture model (instead of the discrete one that is most common now) and derived a relationship between the wire length distribution and the terminal-gate relationship for diamond-shaped regions, under the assumption that this terminal-gate relationship can be represented by a power-law. In [SP86], Sastry et al. used a similar approach and came to the conclusion that, if Rent's rule applies, the wire length distribution must follow a Weibull distribution:

$$\mathcal{N}(\ell) = tp \ell^{p-1} \exp[-tp \ell^p] \quad (3.16)$$

However, their derivation was based on a one-dimensional placement substrate. In [GA89], Gura et al. extended this to a two-dimensional

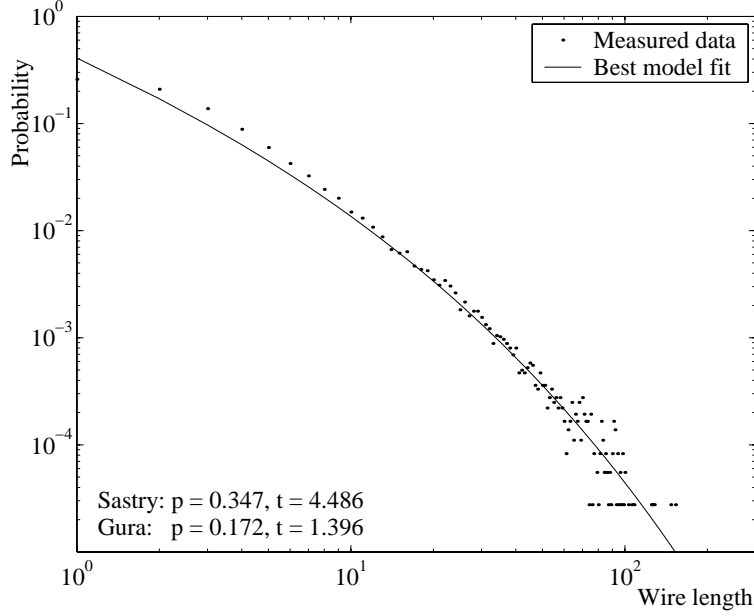


Figure 3.10: Measured wire length for an SA placement of benchmark `ibm01` and fitted distribution resulting from Sastry's and Gura's models.

placement substrate and found:

$$\mathcal{N}(\ell) = tp 2^{p+1} \ell^{2p-1} \exp \left[-2^p t \ell^{2p} \right], \quad (3.17)$$

which is again a Weibull distribution.

For both Sastry's and Gura's versions of this model, t and p are parameters that need to be extracted by fitting the model to the experimental wire length distribution. Also, when choosing the appropriate parameter values for each model, they can yield the same distribution. They should be interpreted as the Rent parameters for one-dimensional and diamond-shaped two-dimensional layout regions, respectively. Note that, since t is present in the exponential term, it is more than merely a scaling constant! We have fitted both models to the experimentally measured wire length distribution for a SA placement of benchmark `ibm01`. The fitted distribution and the corresponding parameter values for each model are shown in Figure 3.10. The resulting values of t and p are very far away from the values we measured for the same benchmark by recursive circuit partitioning.

It is our impression that there are some mistakes in Gura's extension

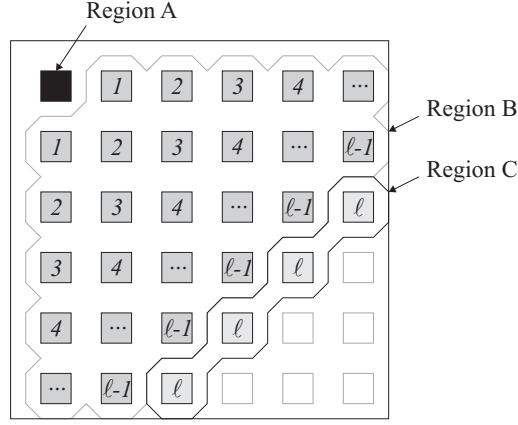


Figure 3.11: Regions *A*, *B* and *C* for the first gate when traversing the layout grid in Davis' technique.

of Sastry's model. In fact, the extension to two dimensions is somewhat more complicated than the way they present it, but we will not go into detail, because that would lead us too far.

3.3.2 Davis' concept of terminal conservation

The basis for Davis' approach is the concept of *terminal conservation* [DDM98], applied to a discrete (gridded) placement architecture, which was originally constrained to be square and have square cells. To explain this, let's consider the three disjoint layout regions *A*, *B* and *C* of Figure 3.11. All gates in region *C* are located at a constant distance ℓ from the single gate location in region *A*. What we would like to derive is the number of nets $N_{A \rightarrow C}$ connecting regions *A* and *C*, because it equals the number of nets $N_A(\ell)$ of length ℓ , connected to a gate located in region *A*. We will denote the terminals of any region *XX* as T_{XX} . Hence the number of terminals of region *A* is T_A and the number of terminals of the region that consists of the union of *A* and *B* is T_{AB} . Also, the number of terminals of, e.g., region *A* that are connected to, e.g., region *C* is indicated as $T_{A \rightarrow C}$. This number equals the number of terminals of region *C* that are connected to *A*: $T_{C \rightarrow A} = T_{A \rightarrow C}$. Indeed, since we only consider two-terminal nets, each net that connects both regions has one terminal that is counted in $T_{A \rightarrow C}$ and one that is counted in $T_{C \rightarrow A}$. Using this notation, we can write down the following

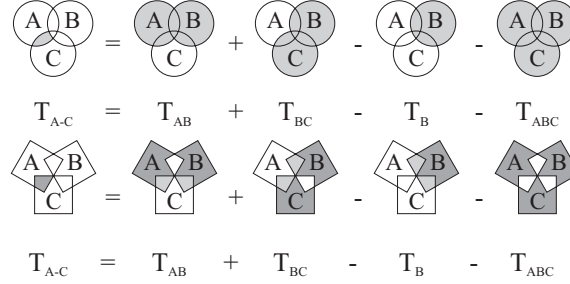


Figure 3.12: Illustration of the concept of terminal conservation: (top) graphical diagram as it was originally used; (bottom) diagram that is mathematically more correct.

equations:

$$T_A + T_B = T_{A \rightarrow B} + T_{B \rightarrow A} + T_{AB} \quad (3.18)$$

$$T_B + T_C = T_{B \rightarrow C} + T_{C \rightarrow B} + T_{BC} \quad (3.19)$$

$$T_A + T_B + T_C = T_{A \rightarrow B} + T_{A \rightarrow C} + T_{B \rightarrow A} + T_{B \rightarrow C} + T_{C \rightarrow A} + T_{C \rightarrow B} + T_{ABC} \quad (3.20)$$

The last term in each expression groups all terminals that connect gates in the respective subregions to gates external to the union of those regions. Combining Equations (3.18), (3.19) and (3.20), we find:

$$T_{A \rightarrow C} + T_{C \rightarrow A} = T_{AB} - T_B + T_{BC} - T_{ABC} = 2N_A(\ell). \quad (3.21)$$

This concept is illustrated graphically in figure 3.12. The top figure is the representation that was used in [CS00, Chr01], while the bottom one corresponds better with the mathematical reality. Indeed, there can be no intersection between regions A , B and C , since this would require three-terminal nets. Also, as in the notation, we have distinguished between both terminals of each net connecting a module pair. Hence, e.g., T_{AB} contains only half the terminals in the intersection regions with C ($T_{A \rightarrow C}$ and $T_{B \rightarrow C}$ but not $T_{C \rightarrow A}$ and $T_{C \rightarrow B}$).

Davis derived the terminal numbers in Equation 3.21 from Rent's rule, using the number of gates in each region (notation G_{XX} , for region XX), leading to:

$$\begin{aligned} N_A(\ell) &= \frac{1}{2}t [(G_{AB})^p - (G_B)^p + (G_{BC})^p - (G_{ABC})^p] \\ &= \frac{1}{2}t [(G_B + 1)^p - (G_B)^p + (G_{BC})^p - (G_{BC} + 1)^p]. \end{aligned} \quad (3.22)$$

If we for now assume that Rent's rule holds under these conditions, Equation (3.22) can be derived for determining the number of nets of length ℓ that connects any gate location to any layout region. The only thing we need is an analytical or numerical way to determine the number of gate positions in regions B and BC .

For a gate located at coordinates $(x, y) = (j, i)$ (i -th row, j -th column), and a region \mathcal{G} that does not contain this gate position, a local site function $\Phi(i, j, \ell)$ can be defined as the number of gate locations in \mathcal{G} at distance ℓ from position (j, i) . If this function is available, G_B and G_{BC} become simple summations:

$$\begin{aligned} G_B &= \sum_{\zeta=1}^{\ell-1} \Phi(i, j, \zeta) \\ G_{BC} &= \sum_{\zeta=1}^{\ell} \Phi(i, j, \zeta) \end{aligned} \quad (3.23)$$

Combining these expressions and Equation (3.22), we find the length distribution of the wires connecting a gate at location (j, i) to the region \mathcal{G} for which $\Phi(i, j, \ell)$ was computed:

$$\begin{aligned} N_{j,i}(\ell) &= \frac{1}{2}t \left[\left(1 + \sum_{\zeta=1}^{\ell-1} \Phi(i, j, \zeta) \right)^p - \left(\sum_{\zeta=1}^{\ell-1} \Phi(i, j, \zeta) \right)^p \right] \\ &\quad - \frac{1}{2}t \left[\left(1 + \sum_{\zeta=1}^{\ell} \Phi(i, j, \zeta) \right)^p - \left(\sum_{\zeta=1}^{\ell} \Phi(i, j, \zeta) \right)^p \right]. \end{aligned} \quad (3.24)$$

To derive the global wire length distribution, Davis proposed to sum partial local wire length distributions for each gate location. He starts with a corner gate location as region A , counting all wires between that gate and the rest of the circuit (Figure 3.11). To avoid counting nets twice, that gate location is then virtually removed from the layout before proceeding to the next location. Hence, the region \mathcal{G} that is used in calculating the local site functions for the second gate location is the entire placement grid minus the first gate location. In this way each gate location is treated, traversing the layout grid from left to right and from top to bottom.

A derivation of the required numeric expressions for $\Phi(i, j, \ell)$ is given in Appendix C. The function $\Phi(i, j, \ell)$, as it was defined by Davis,

is given by:

$$\begin{aligned}
\Phi(i, j, \ell) = & 2\ell\mu_0(\ell) \\
& -(\ell - Y_g + i)\mu_0(\ell - Y_g + i) \\
& -(\ell - j)\mu_0(\ell - j) \\
& +(\ell - Y_g + i - j)\mu_0(\ell - Y_g + i - j) \\
& -(\ell - Y_g + i - 1)\mu_0(\ell - Y_g + i - 1) \\
& -(\ell - X_g + j)\mu_0(\ell - X_g + j) \\
& +(\ell - Y_g + i - 1 - X_g + j)\mu_0(\ell - Y_g + i - 1 - X_g + j) \quad (3.25)
\end{aligned}$$

In this expression, $\mu_0(x)$ represents the discrete step function which equals one if its argument is strictly positive and zero otherwise:

$$\mu_0(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (3.26)$$

3.3.3 Validation and extensions of Davis' prediction technique

Because Davis' approach is based on layout regions, we need to use Rent parameters extracted from geometric Rent characteristics. In fact, only the Rent exponent p is required, since t is merely a scaling parameter. However, in contrast to the partitioning Rent characteristics, geometric characteristics are not available before placement. Therefore, Davis' technique has been used mostly for predicting wiring properties of classes of circuits (e.g., the class of IBM processors), using *typical values* of the Rent parameters for each class. In this approach it is assumed that the geometric Rent parameters are very similar for different designs in a class and that they will not change significantly throughout several design generations. Figure 3.13 shows a wire length distribution, resulting from Davis' model, for a placement grid of 115×115 square cells and a Rent exponent $p = 0.7$.

The full calculation of Davis' model can become quite slow. Hence, several approximations can be made to the original model. A possible approximation can be found by first assuming an unbounded placement grid and using this assumption to derive an expression for the global occupation probability $\mathcal{Q}(\ell)$ (for this derivation, we refer to [DDM98, CS00]):

$$\mathcal{Q}(\ell) \sim \ell^{2p-4} \quad (3.27)$$

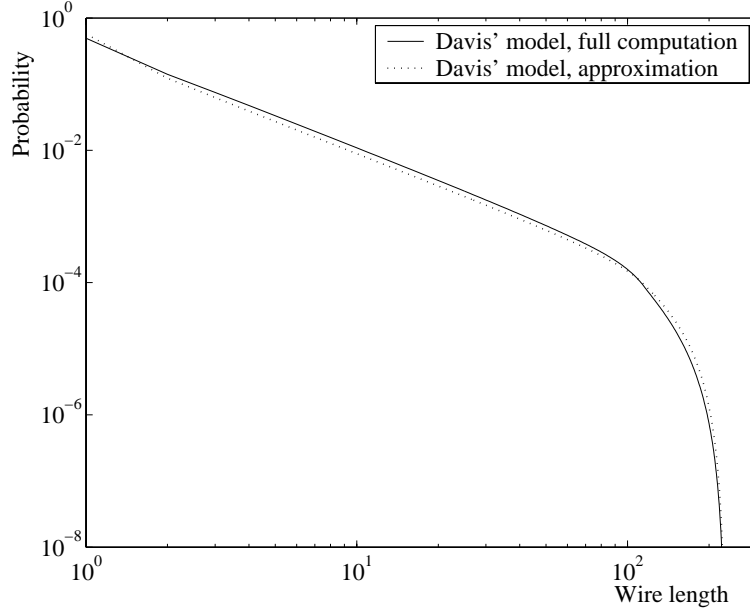


Figure 3.13: Wire length distribution resulting from Davis' model for a placement grid of 115×115 square cells: full model version and approximation.

which is the same expression as that found by Stroobandt (discussed in Section 3.2.3). The global wire length distribution is again found by multiplying this occupation probability with the site function of the placement grid and renormalizing (Equation 3.11):

$$\mathcal{N}(\ell) = \frac{\ell^{2p-4} \mathcal{D}(\ell)}{\sum_{i=1}^{\ell_{max}} i^{2p-4} \mathcal{D}(i)} \quad (3.28)$$

However, in this work, we want to validate the usefulness of Davis' prediction technique for design space exploration. This requires it to be applicable to non-square grids, possible with non-square cells. This can be achieved simply by calculating the local site functions with the techniques presented in Appendix C.

Note that Davis' full model is not symmetrical when applied to non-square grids: the resulting wire length differs depending on whether the grid is traversed row by row or column by column (see Figure 3.14). This is due to the fact that gates are eliminated from the region under consideration as the grid is traversed. To avoid this, it would be better to consider the entire grid for each location of A . This leads us to an

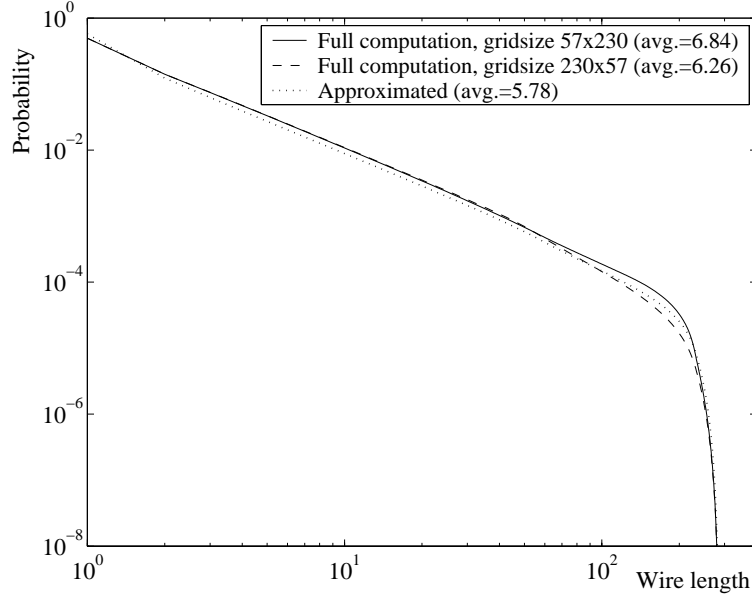


Figure 3.14: Wire length distributions resulting from Davis' full model for grids of 57×230 and 230×57 as well as the approximated model for grids of the same dimensions.

alternative approximation of the model. We propose to use the site distribution to represent the number of gate locations at a distance ℓ from gate location A , *averaged* across possible positions of A . The cumulative of this distribution can then be used to yield the average values for G_B and G_{BC} in Equation 3.22:

$$N_A(\ell) = \frac{1}{2}t [(G_{B,avg} + 1)^p - (G_{B,avg})^p + (G_{BC,avg})^p - (G_{BC,avg} + 1)^p] \quad (3.29)$$

with:

$$\begin{aligned} G_{B,avg} &= \sum_{\zeta=1}^{\ell-1} \mathcal{D}(\zeta) \\ G_{BC,avg} &= \sum_{\zeta=1}^{\ell} \mathcal{D}(\zeta) \end{aligned} \quad (3.30)$$

This approximated version of Davis' model is also shown in Figure 3.13. Obviously, this approximated model does not show the asymmetry that is present in the full model (Figure 3.14).

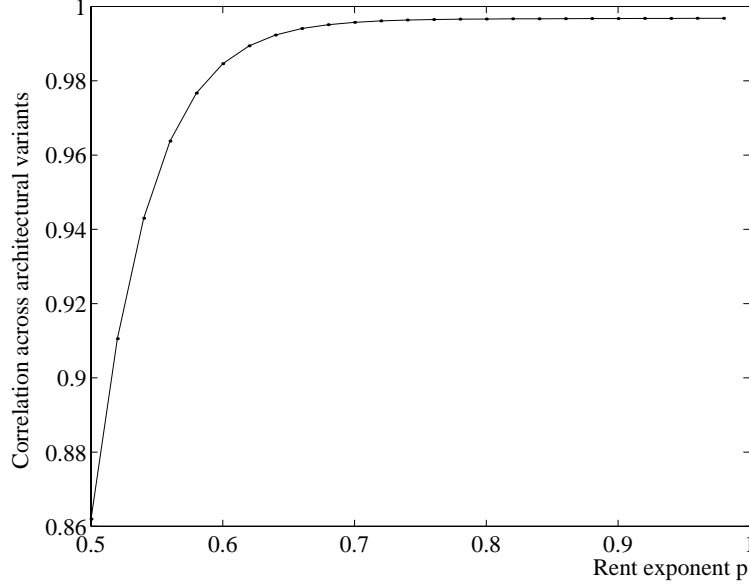


Figure 3.15: Correlation between predicted average wire lengths from Davis' model and experimental results (across 70 architectural variants) for benchmark `ibm17`.

Using this approximation, we can try to find out whether Davis' model can be used for design space exploration. If this is the case, there should be at least one value of the parameter p for each benchmark circuit, for which a good correlation across architectural variants is achieved. If such a value exists, the next step is to determine whether it can be determined *a priori*, i.e. based *only* on the network topology and, if necessary, the placement architecture.

For each of the ISPD98-benchmarks, we have performed placements with `Plato` in grids with aspect ratios of 1 down to 0.1 (decrements of 0.1) and for cells with aspect ratios of: $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, 1, $\frac{4}{3}$, $\frac{3}{2}$, and 2. This comes down to 70 architectural variants for each benchmark. Furthermore, 10 placement runs were performed for each instance and an average was taken across all ten results. The average wire lengths for benchmark `ibm01`, resulting from this experiment, were shown in Figure 2.8. For these cell and grid aspect ratios, the correlation between the predicted average wire lengths from Davis' approximated model and the experimental results was determined for different values of the Rent exponent p . Figure 3.15 shows this correlation (as a function of p) for benchmark `ibm17`. This figure shows that very high correlations

Benchmark	Best value of p	Correlation across architectures
ibm01	0.749	0.946
ibm02	0.829	0.976
ibm03	0.831	0.997
ibm04	0.799	0.968
ibm05	0.887	0.994
ibm06	0.824	0.983
ibm07	0.784	0.996
ibm08	0.815	0.994
ibm09	0.761	0.994
ibm10	0.772	0.997
ibm11	0.765	0.998
ibm12	0.809	0.992
ibm13	0.761	0.994
ibm14	0.772	0.994
ibm15	0.773	0.991
ibm16	0.768	0.989
ibm17	0.785	0.997
ibm18	0.758	0.987

Table 3.2: Rent exponents that yield the best predictions with Davis' model across 70 architectural variants (ISPD98 benchmarks). Also shown is the correlation between measured and predicted values that is achieved with these Rent exponents.

can be achieved.

However, to achieve a high correlation across the different benchmark circuits as well as the architectural variants, the resulting estimates for all benchmarks need to be approximately *as good*. Hence, we have determined the value of p for each benchmark for which the predictions were *on average* closest to the experimental values. The predicted average wire lengths corresponding to those values are shown in Figure 3.16 for some benchmarks. Table 3.2 shows the resulting values of p , as well as the correlations across architectural variants that correspond with them. For all of these benchmarks, the achieved correlations are very high. This seems to indicate that Davis' model might indeed be suitable for design space exploration.

Still, one major problem remains: the values of the Rent exponents were fitted *a posteriori*, i.e. after performing (a large number of) placement experiments. To be useful, we need to be able to measure or es-

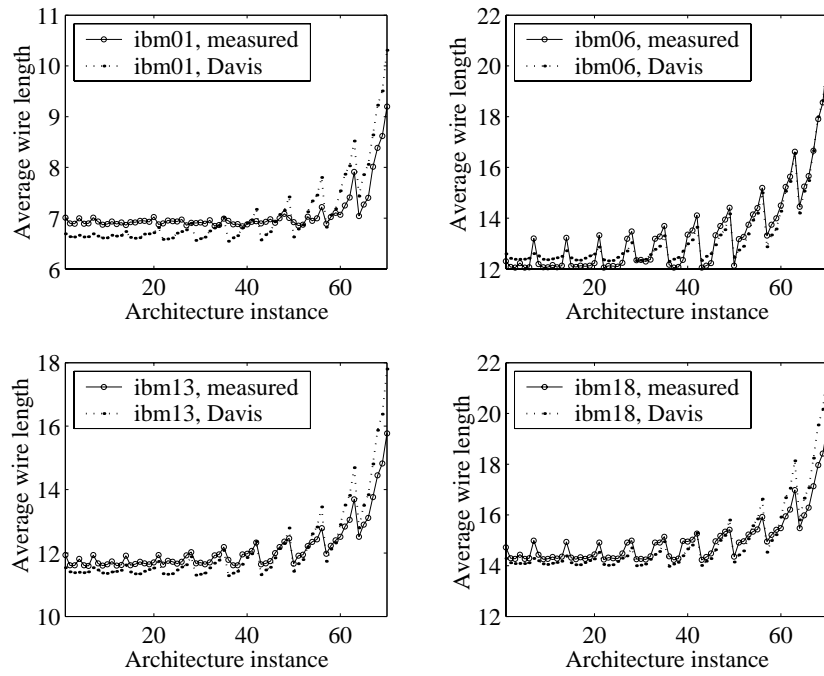


Figure 3.16: Measured average wire lengths and predicted values resulting from using the *best* value of p (i.e., the one for which the predictions were *on average* closest to the experimental values).

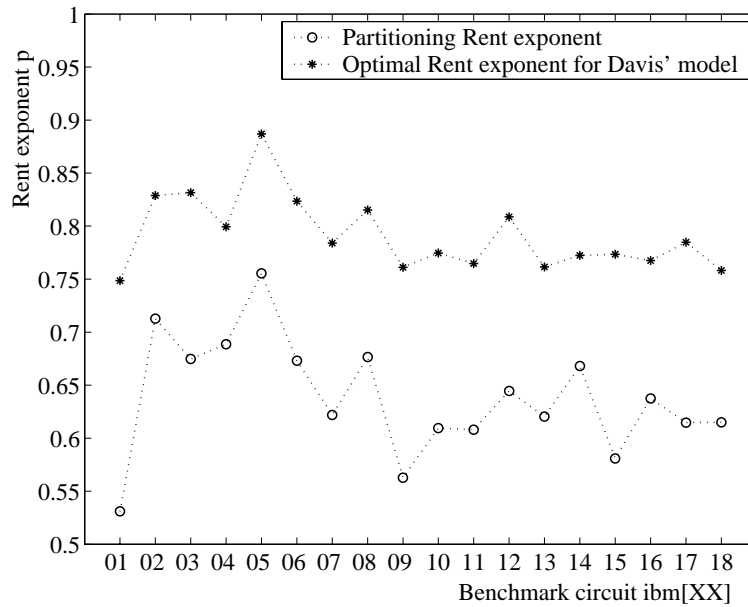


Figure 3.17: Comparison between the Rent exponents used in Donath's model and the fitted Rent exponents that optimize the accuracy of Davis' model.

estimate these values from the circuit netlist. In Figure 3.17, the fitted values are compared to the partitioning Rent exponents used in Donath's model. Clearly, they are not the same! Although there seems to be some correlation between both parameters, the Rent exponents needed in Davis' model are much larger than those extracted by netlist partitioning. Furthermore, experiments have indicated that the results from Davis' model are very sensitive to the Rent exponent value that is used.

3.4 Comparative discussion

Even with the extensions proposed in [VM00], Donath's technique for predicting wire length distributions remains restricted to placement grids for which the number of cells in all directions are powers of two. Even when a scaling is applied to obtain average wire lengths (as we did in our experimental validation for industrial benchmarks), the possible grid aspect ratios are still greatly restricted.

In contrast, Davis' geometric prediction technique seems to be able to yield relatively accurate predictions, with good correlations across a wide range of grid and cell aspect ratios. However, only very few indications can be found in literature with regard to the Rent exponents that have to be used. In fact, to our knowledge, the only publication that links netlist properties to geometric Rent exponents is [Chr00]. In this paper, Christie proposes a model that links the placement Rent exponent to the average number of terminals per gate t_G and the recursive partitioning strategy (four-way vs. bipartitioning). However, we have already determined that the circuit partitioning properties (expressed by, e.g., the partitioning Rent exponent) are also of major importance (Section 2.5.3, Figure 2.10). Another issue with Davis' model is the fact that *the same Rent parameters* are used for a very wide range of regions (and region shapes) when traversing the placement grid.

The partitioning Rent parameters used in Donath's technique, at least, can be measured for each individual circuit. However, to obtain them, an entire recursive partitioning must be performed. This can be quite time-consuming, making the additional cost of performing an actual partitioning based placement with some additional greedy swapping optimization almost negligible. This almost excludes a priori interconnect prediction techniques for some applications. Are there no faster ways of obtaining the terminal-gate relationship resulting from recursive partitioning?

Finally, our experiments show that even the small deviations from Rent's rule in the synthetic benchmarks caused a difference between the predicted and experimental average wire lengths (Table 3.1). This suggests the far more extensive deviations from Rent's rule found in real circuits as a possible cause for the bad correlation between predicted and experimental average wire length we found for those circuits. A similar argument applies to Davis' technique, where we have already mentioned that the results are very sensitive to the Rent expo-

	Donath-type	Davis-type
Architectural flexibility:		
Grid dimensions:	\mp	++
Cell dimensions:	\mp	++
Topological properties:	\pm	??

Table 3.3: Comparison between Donath-type models and Davis-type models with respect to their ability to model the impact of the placement architecture and the circuit topology.

ment that is used. Since Rent's rule is only a very rough approximation for partitioning-based Rent characteristics, we can expect that this is also the case for geometric Rent characteristics. Hence, this model too might benefit from better models for the terminal-gate relationship.

Table 3.3 summarizes the performance of topological (Donath-based) and geometrical (Davis-based) approaches to a priori interconnect prediction, including the extensions to the original models. So which way do we need to go to obtain an a priori interconnect prediction technique that provides sufficient sensitivity to variations in the circuit topology as well as the placement architecture? Is any of the standard approaches even useful? If the questions regarding the proper Rent parameter values can be solved, the geometric approach seems very promising. However, if an approach, similar to Donath's can be derived that offers full flexibility with respect to the placement architecture, it might turn out to be the best candidate.

In the next chapter, we study topological as well as geometrical terminal gate-relationships and try to answer some of the questions that have risen in this section. Based on this discussion, we will decide on a prediction approach that best meets our requirements.

Chapter 4

Comparison of different terminal-gate relationships

In the first section of this chapter, we have a closer look at different types of module generation strategies and the Rent characteristics resulting from them. Then, in Section 4.2, we focus on Rent characteristics that are extracted by recursively partitioning the circuit. We discuss and adapt a model for recursive partitioning that was originally proposed by Verplaetse [Ver03] and show how it can be used to predict the entire partitioning terminal-gate relationship by performing only a very limited number of partitioning levels. Finally, we study Rent characteristics that are generated by “cutting” regions from a circuit layout (geometric Rent characteristics) in Section 4.3. Starting from an analytical model proposed by Christie [Chr01], we move to ever more detailed models in an attempt to capture and explain our observations on this type of terminal-gate relationships. We end up with a numerical model that considers them as a consequence instead of the cause of the wire length distribution.

4.1 Types of terminal-gate relationships

From the previous chapter, we know that the basis for most current interconnect prediction techniques is the relationship between the average number of terminals required to connect a subset of the gates in a circuit (a *module*) to the rest of the circuit and its exterior.

In the past, when used in interconnect prediction, these terminal-gate relationships have been modeled by Rent’s rule, given by Equation (2.2). However, it has always been known that this approximation can

offer only limited accuracy, with substantial deviations occurring for large, as well as for small module sizes. When applying Donath's technique for interconnect prediction (Section 3.2.2), these deviations were identified as the most probable cause of the lack of correlation between model and experimental results. For geometric prediction techniques, the situation was even worse, in that we could not even determine *a priori* which Rent parameters to use in the models.

Hence, if we want to improve the accuracy of the existing interconnect prediction techniques, a first step is to examine these terminal-gate relationships more closely. That is the main intention in this rather theoretical chapter. For each of the two fundamental approaches to predicting wire length distributions, the partitioning-based approach and the geometric approach, we try to answer the following questions:

- What is the appropriate Rent characteristic to be used?
- What does it really look like if we measure it?
- Can we derive a mathematical model that fits these empirical data?
- Can the parameters in this model be determined *a priori*?

The typical features of a Rent characteristic show up for a very wide range of module generation strategies. This is even the case for random modules [Ver03]. Indeed, on average, the gates in a module that consists of G randomly selected blocks will have

$$t_G G - \frac{G}{G_c} T_{ext} = t'_G G \quad (4.1)$$

terminals to connect them to other gates (if there are only two-terminal nets). For each of these connections, there are $(G_c - 1)$ gates they can be connected to. Only $G_c - G$ of those possible gates are external to the module. Hence, on average, for any module gate terminal the probability that it is an external terminal for the module is:

$$P_{ext} = \frac{G_c - G}{G_c - 1}. \quad (4.2)$$

Remember that this concerns the gate terminals that are *not* external for the circuit. Obviously, gate terminals that are external for the circuit are

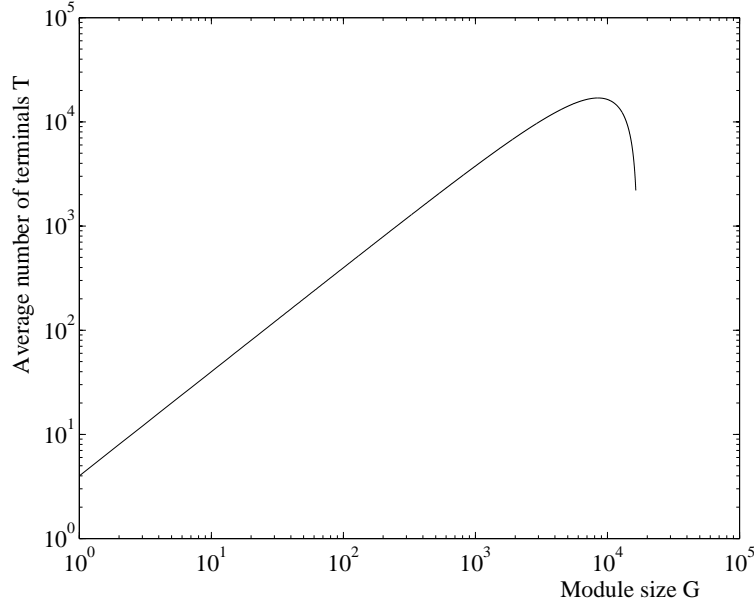


Figure 4.1: Expression (4.3) for the average terminal-gate relationship for random clusters (using the values of t'_G , G_c and T_{ext} corresponding to benchmark q65).

also external for the module, yielding a total expected average number of module terminals of

$$T(G) = t'_G G \frac{G_c - G}{G_c - 1} + \frac{G}{G_c} T_{ext}. \quad (4.3)$$

This expression is shown in figure 4.1.

However, in most graphs there exists some degree of locality. This means that modules can be generated for which many more connections are internal than predicted by the random cluster model. Some module generation strategies actively search for these modules. This is certainly the case for a recursive bipartitioning that minimizes the number of nets cut at each level. In other cases, locality is exploited to minimize a different cost function. For instance, circuit placement tries to place connected gates as close to each other as possible to keep wire lengths short. As a result, modules originating from connected regions in a circuit layout also tend to have a lower number of terminals than random clusters. In every application of Rent's rule for interconnect prediction, it was understood that the modules under consideration were the results of some such terminal minimization. However, the

constraints regarding which modules were allowable were not always very strict.

In general, we can make the distinction between *topological circuit modules*, i.e., modules that are derived in some way directly from the circuit topology, and *geometric circuit modules*, i.e., modules that were generated by cutting a coherent region from a circuit layout. For this last type, we can expect the number of terminals for these modules, and hence the resulting Rent characteristic, to depend on the placement algorithm and the placement quality.

Besides the division into topological and geometric module generation strategies, one can also distinguish between recursive partitioning strategies and module growth strategies. In the first case, for each module size, the number of terminals is the average across a number of (approximately) equally sized, disjoint modules. Together, these modules cover the entire circuit graph. The partitioning Rent characteristic, which is obtained from the circuit topology by recursive bipartitioning, falls into this category. Indeed, it is characteristic for a partitioning tree that at each hierarchy level the modules are disjoint, in the sense that every gate is contained in exactly one module. Recursive clustering is another topological member of this category. Also, a partitioning can be created geometrically by, e.g., applying a regular *partitioning grid* to the layout to divide it into modules.

In contrast, in module growth strategies, modules of a given size are created from a randomized seed gate or cluster of gates. Then, incrementally, some gates are added to that module according to a specified criterion. This process is repeated for many (or all) different seeds. The average is then taken across all modules of equal size, originating from different seeds. As a result, modules of equal size are not necessarily disjoint. When growth modules are generated in a topological way, the most common criteria are based on adding one or multiple gates that are directly connected to gates already in the module. The order in which these gates are added can, for instance, be based on how strongly (i.e., by how many edges) they are connected to the module.

For growth modules extracted from a layout, the most common strategy is to “grow” a region with a given shape, e.g., rectangular regions with a constant *aspect ratio* χ ($\chi \in]0, 1]$), or diamond-shaped regions. The initial rectangular region can, e.g., be considered as a corner gate for the region. Then, at each step, gates are added to create a larger region, with the same shape and the same corner gate. For

Table 4.1: Rent parameters resulting from different module generation strategies, extracted by fitting Rent’s rule to the Rent characteristics in Figure 4.2

Topological module generation method	t	p
Edge clustering	9.27	0.599
Neighbor clustering	2.91	0.914
Random partitioning	4.15	0.985
Recursive balanced partitioning (hMetis)	3.94	0.607
Recursive balanced partitioning (FM)	4.03	0.606

Geometric module generation method	t	p
Average layout (random placement)	6.02	0.991
Average layout (SA), square regions	10.92	0.648
Average layout (SA), region aspect ratio $\chi = 1/5$	10.52	0.689
Layout partitioning (SA), square regions	8.93	0.657

diamond-shaped regions, the first gate can be the central gate of the diamond. Here, each step corresponds to a distance ℓ with respect to that central gate and all gates are added that are no more than a distance ℓ apart from it. For both rectangular and diamond-shaped regions, the same modules can also be found by randomly positioning a rectangular or diamond shaped mask on the circuit layout and selecting all gates within the mask as module gates. Again, many other module growth strategies can be formulated.

In short, both topological and geometric modules can be created in many different ways, each resulting in a different Rent characteristic and sometimes quite different values of the derived Rent parameters. Figure 4.2 shows some topological and geometric Rent characteristics for the ISPD98 benchmark circuit `ibm01`, resulting from different module generation approaches. All curves in these figures show the typical features of a Rent characteristic: they more or less follow a power law, usually with deviations for small and large module sizes. However, the corresponding values of the parameters t and p (Table 4.1) are sometimes quite far apart.

From these observations, we can conclude that the concept of a Rent characteristic is inseparably linked to the way in which the modules were created and used in averaging. From a theoretical point of view,

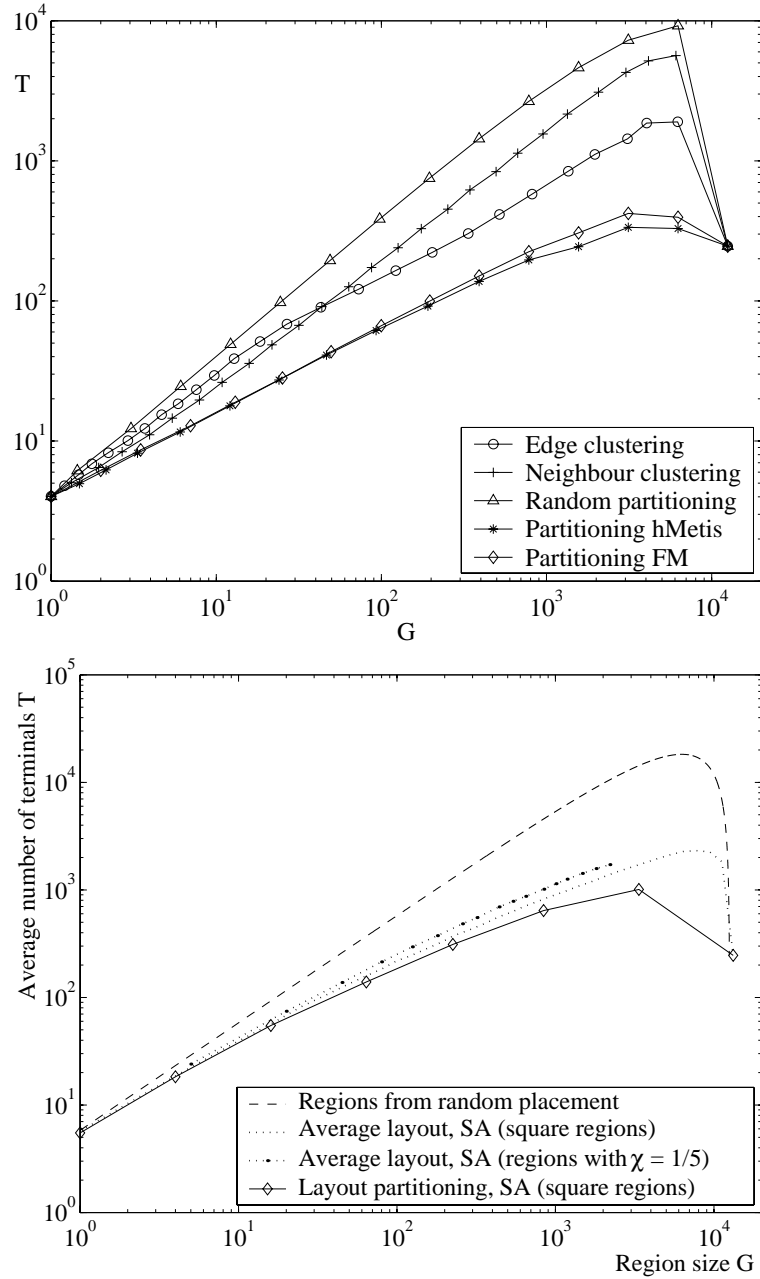


Figure 4.2: Rent characteristics resulting from different module generation strategies for *ispd98* benchmark *ibm01*: (top) topological Rent characteristics; (bottom) geometric Rent characteristics.

Table 4.2: Types of Rent characteristics considered in this chapter

	Topological	Geometric
Partitioning	$T_{(p)}(G)$ Graph (bi)partitioning (recursive)	—
Growth	—	$T_{(al)}(G)$ Average layout (fixed region shape)

it would be interesting to try and derive probabilistic models for many of these Rent characteristics. However, in the rest of this chapter, we will concentrate on those types of Rent characteristics that are currently important for the interconnect prediction techniques described in the previous chapter.

Therefore, in the category of topological module generation strategies, we restrict our discussion to recursive optimized bipartitioning. Partitioning trees can be generated top-down, by actual hierarchical partitioning, but also bottom-up, by hierarchical clustering. Also, different optimization criteria and heuristics can be used. The partitioning or clustering algorithm that is used turns out to affect the resulting Rent characteristic. In fact, in [HKKR94], this fact was used as the basis for comparing different partitioning algorithms. In the rest of this chapter, the term *(topological) partitioning Rent characteristic* and the notation $T_{(p)}(G)$ are used exclusively to indicate topological Rent characteristics, resulting from recursive bipartitioning with the well-known academic partitioning tool `hMetis` [KK98]. We will not consider any topological Rent characteristics resulting from module growth strategies. For geometric Rent characteristics, we focus on randomly selecting modules of a given shape from the layout (growth-based strategy). We will refer to this type as *average layout Rent characteristics*. In our symbolic notations, we use the index *al*, i.e. $T_{(al)}(G)$. Rent characteristics resulting from a recursive partitioning of the circuit layout (i.e., *layout partitioning Rent characteristics*) are discussed in [VDSVC01, DVSVC03].

4.2 Circuit partitioning

4.2.1 The partitioning Rent characteristic

Figure 4.3 shows the partitioning Rent characteristic $T_{(p)}(G)$ for benchmark `ibm17`. It can be obtained from the circuit netlist by performing a full recursive partitioning. However, since each bipartitioning step is in itself a hard optimization problem, this type of measurement may take quite a lot of time. In many partitioning-based placement tools, the creation of a partitioning tree takes up a very large fraction of the total placement time. Hence, if a full recursive partitioning is performed, we are not far from performing the type of constructive wire length prediction presented in [KS01, YCS02]. To reduce the computation time that is required for estimating the Rent exponent, even early work suggested to use only a limited number of partitioning levels. For instance, in [HKKR94], recursive partitioning stopped as soon as the module sizes were below a given threshold size. The resulting data points, excluding values belonging to region II, were used for fitting Rent's rule.

We have evaluated this approach, using a fixed number of partitioning levels instead of a module size threshold. The T -values corresponding the other partitioning levels were obtained by interpolation in the logarithmic domain. Figure 4.3 shows an example for benchmark `ibm17`, with only 5 partitioning levels used for the interpolated Rent characteristic. We have evaluated the difference between the measured and the interpolated Rent characteristic using the following error function:

$$E_T = \frac{\sum_{i=1}^{I_{max}} [\log_2(T_i) - \log_2(T_{i,fit})]^2}{I_{max}}. \quad (4.4)$$

In Table 4.3, these error values are given for 3, 6 and 9 partitioning levels and for benchmarks `ibm01` - `ibm18`.

It turns out that a relatively large number of partitioning levels is required to obtain a good fit. However, these results were obtained by performing *blind* interpolation in the (partial) Rent characteristic, i.e., no additional information about the shape of the Rent characteristic was used. We can expect that better results can be achieved if a model for the Rent characteristic is available.

What we would really like is an accurate parametric model, with a limited number of parameters that can be easily and rapidly calculated from the circuit topology, without the need for partitioning. Unfortunately, such a model has not yet been found. However, in the next sec-

Table 4.3: Errors E_T for interpolating the Rent characteristic using 3, 6 and 9 partitioning levels, respectively.

benchmark	3 levels	6 levels	9 levels
ibm01	0.138	0.0220	0.00155
ibm02	0.192	0.00549	0.000821
ibm03	0.0487	0.0102	0.00222
ibm04	0.0992	0.00659	0.00159
ibm05	0.0413	0.00185	0.000185
ibm06	0.0785	0.00671	0.000451
ibm07	0.105	0.0133	0.000774
ibm08	0.140	0.0183	0.00163
ibm09	0.211	0.0522	0.00605
ibm10	0.209	0.0290	0.00238
ibm11	0.159	0.0318	0.00325
ibm12	0.175	0.0427	0.00223
ibm13	0.477	0.0557	0.00696
ibm14	0.283	0.0361	0.00142
ibm15	0.231	0.0767	0.0167
ibm16	0.386	0.0513	0.00798
ibm17	0.363	0.0948	0.00940
ibm18	0.433	0.0879	0.00455
Average	0.209	0.0357	0.00390

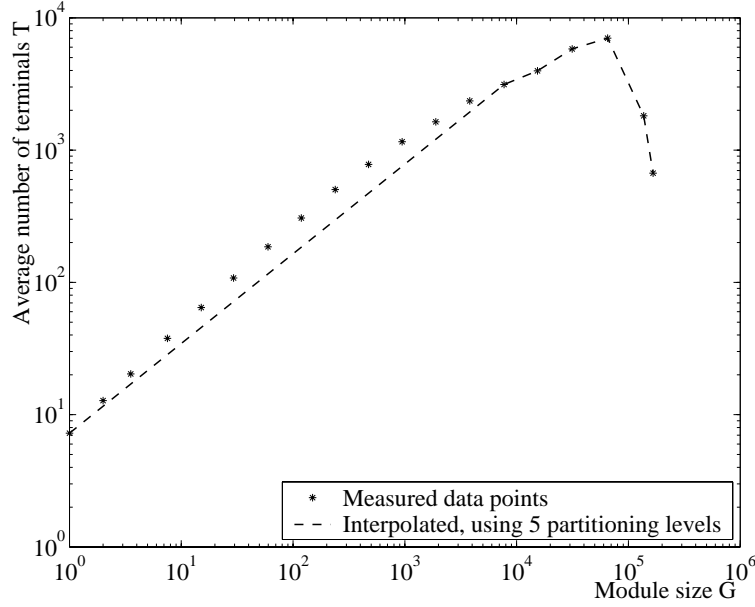


Figure 4.3: Partitioning Rent characteristic for benchmark `ibm17`: measured data points and interpolation using 5 partitioning levels.

tion, we describe a model for recursive graph bipartitioning that allows very accurate estimations of the complete Rent characteristic, based on only a few levels of recursive partitioning.

4.2.2 Verplaetse's recursive graph bipartitioning model

Recursive partitioning equations

In his Ph.D. thesis [Ver03], Verplaetse studied the partitioning properties of circuit graphs and derived several models (of increasing accuracy) for Rent characteristics resulting from optimized hierarchical circuit partitioning. As in Donath's technique, the partitioning levels are indexed from H , the top level or the entire circuit graph, to 0 or the lowest level, where all modules consist of a single block.¹ Since only perfectly balanced partitionings are considered, the graph size neces-

¹Note that Verplaetse indexed the hierarchy levels in the reverse order, i.e., the entire circuit is level 0, while a single gate corresponds to level H . To avoid confusion, we have chosen to use the same indexing scheme as that used in our description of Donath's prediction technique.

sarily is a power of 2, i.e., $G_c = 2^H$. At a given hierarchy level k , the module size is given by:

$$G_k = 2^k. \quad (4.5)$$

The average number of terminals (or external nets) for a module at that level is indicated by T_k , while the average number of internal nets per module is N_k .

The basis for Verplaetse's models is a set of recursive equations. Although he also proposed a model that includes multi-terminal nets, we restrict the present discussion to circuits with two-terminal nets only. In this case, the number of terminals T_k , external to a module at a given hierarchy level k , always equals half the number of terminals T_{k+1} at the level above, incremented by the additional external terminals that were created by partitioning (see Figure 4.4):

$$\begin{aligned} T_k &= \frac{T_{k+1}}{2} + N_{cut,(k+1)} \\ &= \frac{T_{k+1}}{2} + \alpha_{k+1} N_{k+1}. \end{aligned} \quad (4.6)$$

In this expression, α_{k+1} denotes the fraction of the nets at level $k+1$ that were cut during partitioning (the *cut probability*). The number of internal nets at level k equals half of the nets that were not cut at the previous level (also shown in Figure 4.4):

$$N_k = \frac{(1 - \alpha_{k+1})}{2} N_{k+1} \quad (4.7)$$

First, Verplaetse considers balanced random bipartitioning. For each two-terminal net, there are four possible assignments of the two gates g_1 and g_2 connected to it to the two modules. Only for the two assignments where both are in different modules, the net is cut. The probability for the first gate to be assigned to the first module equals

$$\frac{G_k/2}{G_k}.$$

With the first gate assigned, there remain $G_k - 1$ "slots" for assigning the second gate, resulting in a probability

$$\frac{G_k/2}{G_k - 1}$$

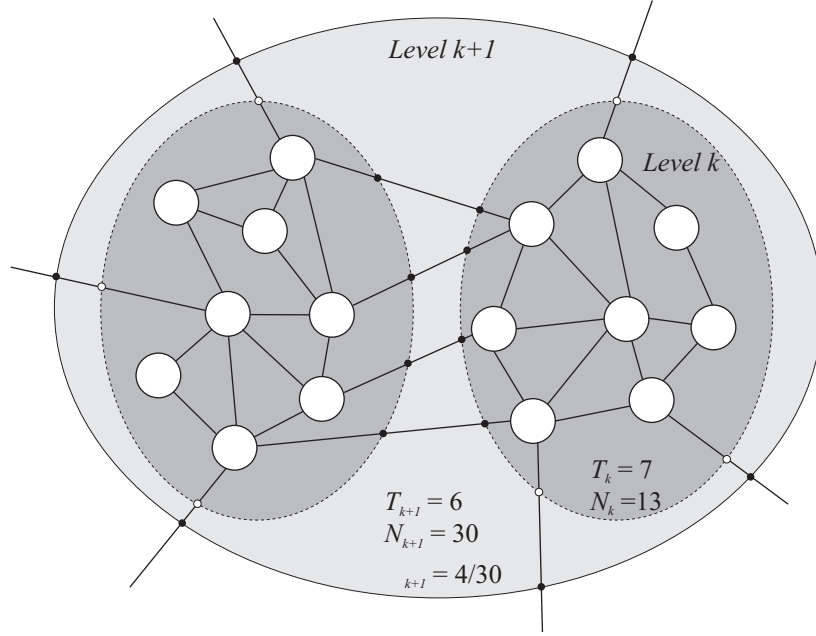


Figure 4.4: Graphical representation of a balanced bipartitioning step in Verplaetse's recursive equations. The module at level $k + 1$ has 6 terminals, 30 internal nets and it contains 16 gates. The number of nets cut at that level is 4, so $\alpha_{k+1} = 4/30$. As a result, the modules at level k on average have 7 terminals and 13 internal nets.

of assigning the second gate to the second module. The same reasoning applies for the reverse assignment. As a result, for any individual net at level k , the probability that the blocks connected to it are in different modules equals:

$$\alpha_{k,rand} = 2 \frac{G_k/2}{G_k} \cdot \frac{G_k/2}{G_k - 1} = \frac{1}{2} + \frac{1}{2(G_k - 1)} \quad (4.8)$$

The combination of Equations (4.6) and (4.7), using Expression (4.8) for α_k , can be solved to [Ver03]:

$$T_k = \left(t_G - \frac{T_{ext}}{G_c} \right) G_k \left(\frac{G_c - G_k}{G_c - 1} \right) + \frac{T_{ext}}{G_c} G_k. \quad (4.9)$$

This equation is in line with the number of terminals for random circuit modules, given by Equation (4.3).

This simple exercise illustrates that even random recursive partitioning results in a Rent characteristic that is close to a power law (with

unit slope) for a broad range of module sizes (see Figure 4.8 for an example). Also, a second region is present in this model, and the values for $T_0 = T_{ext}$ and $T_H = t_G$ are correct.

Obviously, for optimized bipartitioning, the α_k should be smaller than in the random case. Verplaetse identifies two boundary conditions that must be met. First, for a sufficiently large number of blocks in the circuit, the top level value of α for random partitioning approximately equals 0.5. Hence, approximately, $\alpha_0 \leq 0.5$ for optimized partitioning. Second, at level $k = 1$, all remaining internal nets are cut, resulting in $\alpha_1 = 1$.

Note that for a circuit that obeys Rent's rule exactly, we would have:

$$\begin{cases} \alpha_k N_k &= \frac{2T_{k-1} - T_k}{2} \\ N_k &= \frac{tG_k - T_k}{2} \end{cases}, \quad (4.10)$$

and hence:

$$\alpha_k = \frac{2T_{k-1} - T_k}{tG_k - T_k} = \frac{2^{1-p} - 1}{G_k^{1-p} - 1}. \quad (4.11)$$

This expression does fulfill the boundary condition at $k = 1$, but offers no flexibility regarding the number of external connections T_{ext} .

Cut probabilities from experimental data

To directly obtain the α_k -values, corresponding to experimentally measured terminal-gate relationships, we propose the following alternative to Equation (4.7):

$$N_k = \frac{tG_k - T_k}{2}. \quad (4.12)$$

This equation expresses the number of nets at level k as a function of the total number of gate terminals at that level and allows the elimination of N_{k+1} from Equation (4.6). The result is a single recursive equation:²

$$T_k = \frac{(1 - \alpha_{k+1})}{2} T_{k+1} + \frac{\alpha_{k+1}}{2} tG_{k+1}$$

²This recursive equation can be solved into the following closed expression:

$$T_k = 2^{k-H} \left[\prod_{i=k+1}^H (1 - \alpha_i) \right] \left[t_G G_c \left[\sum_{i=k+1}^H \frac{\alpha_i}{\prod_{j=i}^H (1 - \alpha_j)} \right] \right], \quad (4.13)$$

which, in its general form, is extremely impractical. Furthermore, taking into account the range of possible values for α_k ($[0, 1]$), there exist, to our knowledge, no simple analytical approximations for this expression.

$$= \frac{1}{2} [T_{k+1} + (t_G G_{k+1} - T_{k+1}) \alpha_{k+1}], k = 0 \dots (H-1). \quad (4.14)$$

Solving this equation for α_k , we find:

$$\alpha_k = \frac{2T_{k-1} - T_k}{t_G G_k - T_k}, k = 1 \dots H. \quad (4.15)$$

Strictly speaking, Verplaetse's recursive equations only apply to recursive circuit partitionings that result in a perfectly balanced partitioning tree. In that case, the circuit size, as well as all G_k -values, are equal to powers of two. This restriction is usually not fulfilled for the data points of real, experimentally measured Rent characteristics $T_i(G_i)$ ($i = 0 \dots i_{max}$ and $G_0 = G_c$). Generally, for these values, $G_c \neq 2^H$, $G_i \neq 2^i$, and the G_i are not perfectly equidistant on a logarithmic scale.

Hence, we can not extract values α_k *directly* from a measured partitioning Rent characteristic. Instead, we must first derive a set of (G_k, T_k) values from the measured terminal-gate relationship that *do* represent a balanced bipartitioning tree, but that still reflect the essence of the circuit's partitioning properties.

First, we need to determine the value of H . The obvious choices are either $H = \lfloor \log_2(G_c) \rfloor$ or $H = \lceil \log_2(G_c) \rceil$. In this work, we use the second option.

Second, we transform the data points $T_i(G_i)$ in the Rent characteristic into a new set of data points $\tilde{T}_i(\tilde{G}_i)$. At the top level, this transformation must yield:

$$\begin{cases} \tilde{G}_0 &= 2^H \\ \tilde{T}_0 &= T_0 \end{cases} \quad (4.16)$$

Also, a data point corresponding to $G = 1$ must remain present.

We have achieved this by *stretching* the Rent characteristic in the logarithmic domain. Mathematically, this results in:

$$\begin{cases} \log_2(\tilde{G}_i) &= \log_2(G_i) \frac{H}{\log_2(G_c)} \\ \tilde{T}_i &= T_i \end{cases} \quad (4.17)$$

This transformation is illustrated in Figure 4.5. It leaves the lower point of the Rent characteristic unchanged. Also, if the original partitioning tree was almost balanced, the resulting values of \tilde{G}_i will end up very close to powers of two. This is clearly visible in Figure 4.5, where the powers of two are indicated by vertical dotted lines.³

³Note that Verplaetse's original approach for obtaining the (G_k, T_k) values was slightly different, but yielded very similar results.

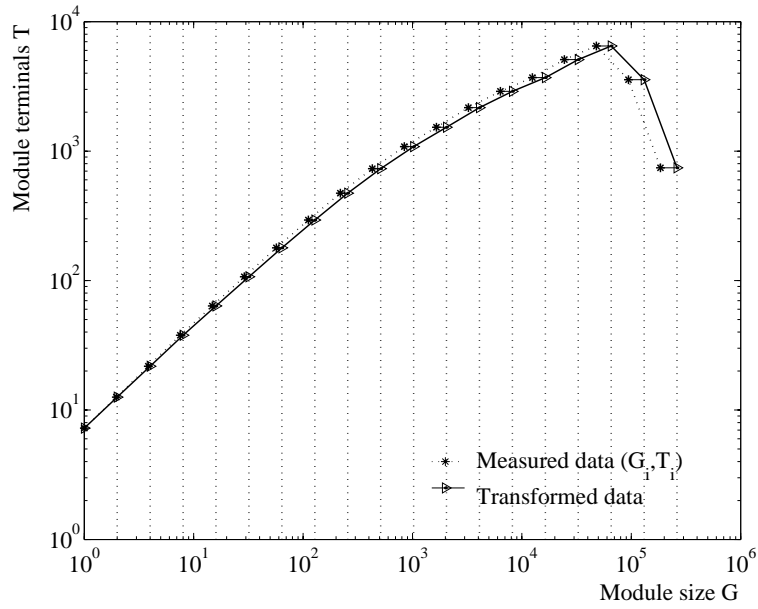


Figure 4.5: Graphical illustration of transformation 4.17, yielding the transformed data points $(\tilde{G}_i, \tilde{T}_i)$. The presence of the vertical grid lines, indicating the powers of 2, shows how close the resulting values \tilde{G}_i are to the target values $G_k = 2^k$.

After transforming the original data points, we can choose $G_k = 2^k$ ($G_H = 2^H$). The T_k -values corresponding to these module sizes can be found by interpolation⁴ in the transformed Rent characteristic $(\tilde{G}_i, \tilde{T}_i)$. Because the \tilde{G}_i are close to the desired G_k , the resulting T_k -values will be very close to the original T_i -values (see Figure 4.5). Figure 4.6 (top graph) shows the resulting α_k corresponding to a recursive optimized bipartitioning of benchmark circuit `ibm17`.

Verplaetse's cut probability models

The question is now whether the resulting α_k can be approximated more accurately than T_k , preferably by a simple analytical function of G_k . Figure 4.6 indicates that this might indeed be the case. The resulting model could then be used to significantly reduce the number of partitioning levels required for interconnect prediction. Such a model would have another very promising property. Indeed, in interconnect prediction techniques that are based on circuit partitioning properties, the numbers of nets cut at each hierarchy level are required. These are derived from the partitioning Rent characteristic. Hence, a direct estimation of these numbers, or of the cut probabilities α_k , can be expected to result in increased accuracy. Therefore, in the remainder of this section, we will evaluate models by their ability to estimate the α_k -values, instead of the T_k -values. For this purpose, we use the error function:

$$E_\alpha = \frac{\sum_{k=0}^{H-1} [\log_2(\alpha_k) - \log_2(\alpha_{k,fit})]^2}{H}, \quad (4.18)$$

instead of the previously introduced error function E_T (4.4).

To derive an expression for α_k , Verplaetse [Ver03] interprets the first part of Equation (4.8) as the fraction of nets that would be cut when randomly partitioning an infinitely large circuit and the second part as a correction caused by the finite circuit size and the demand for a balanced partitioning. Hence, to model optimized partitioning, he proposes to reduce the first term by a constant factor $\tau \leq 1$. This results in a first model for α_k with only that single parameter and a new second term to meet the boundary condition $\alpha_{H-1} = 1$:

$$\alpha_k = \frac{\tau}{2} + \frac{1 - \tau/2}{G_k - 1}, \quad (0 \leq \tau \leq 1). \quad (4.19)$$

⁴We have used linear interpolation in the logarithmic domain, but alternatives are possible.

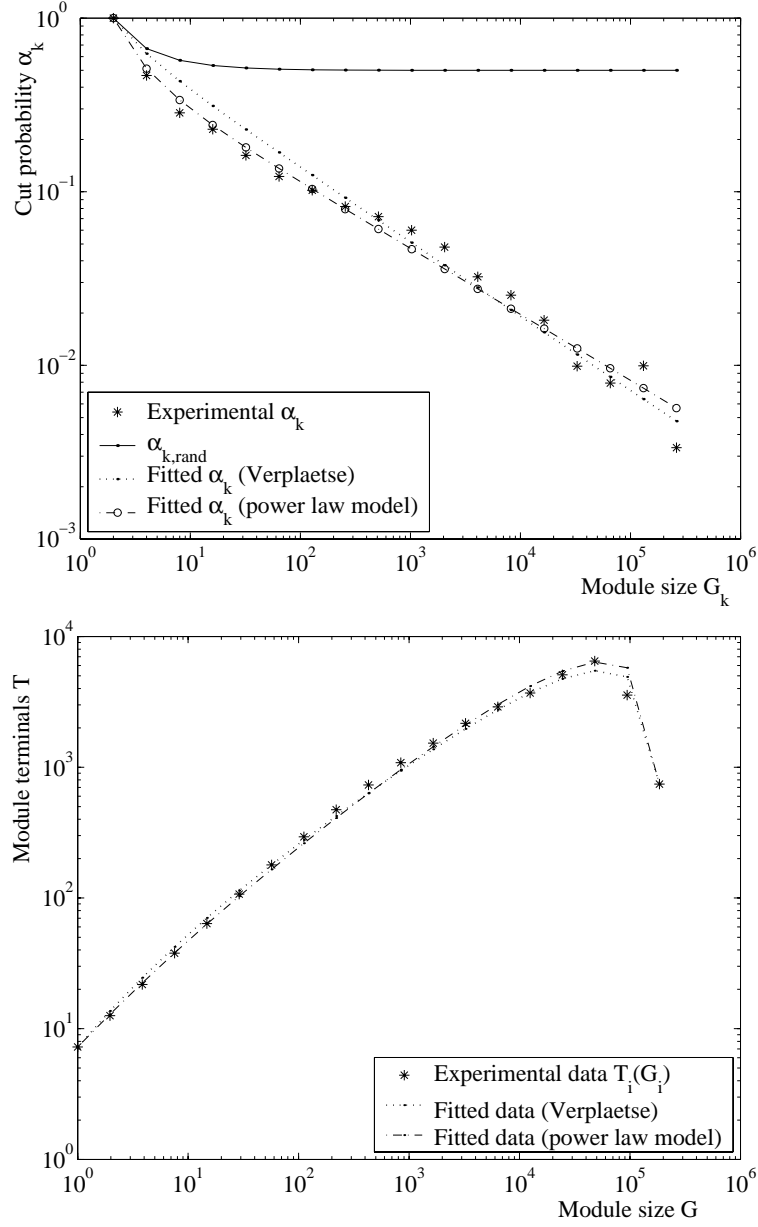


Figure 4.6: (top) Values of α_k corresponding to the recursive bipartitioning of benchmark circuit *ibm17* (with two-terminal nets only): experimental and fitted values resulting from both model versions, as well as the α_k -values corresponding to a random partitioning; (bottom) corresponding values of T_k .

Using this expression for α_k , an analytical expression for T_k is no longer available. Still, a numerical solution of the recursive Equations (4.6) and (4.7) is always possible (see Figure 4.7 for the α_k and T_k resulting from a range of τ -values). Verplaetse found that the resulting Rent characteristics are very similar to those resulting from optimally partitioned random graphs. For real circuit graphs, however, the Rent characteristic is usually straight or convex (on a logarithmic scale), rather than concave, for small module sizes. Hence, Verplaetse proposed a two-parameter variant to correct this:

$$\alpha_k = \frac{\tau}{2} + \frac{1 - \tau/2}{(G_k - 1)^\epsilon}, (0 \leq \tau \leq 1). \quad (4.20)$$

For small values of ϵ (sufficiently < 1), this results in convex Rent characteristics.

Alternative cut probability model

Independently, we have derived an alternative expression for α_k . Our approach was to compare the cut probabilities obtained from optimized recursive bipartitioning to those given by Equation (4.8), which correspond to random recursive bipartitioning. Figure 4.9 illustrates the relation between both sets of cut probabilities for benchmark circuit `ibm17`. We found that it can be approximated quite accurately by a simple power law:

$$\begin{aligned} \alpha_k &= CG_k^{-\beta} \alpha_{k,rand} \\ &= CG_k^{-\beta} \frac{G_k}{2(G_k - 1)}. \end{aligned} \quad (4.21)$$

In this expression, C is a constant value that can be determined from the boundary condition $\alpha_{H-1} = 1$:

$$C = 2^\beta, \quad (4.22)$$

resulting in

$$\alpha_k = \frac{(G_k/2)^{-(\beta-1)}}{G_k - 1}. \quad (4.23)$$

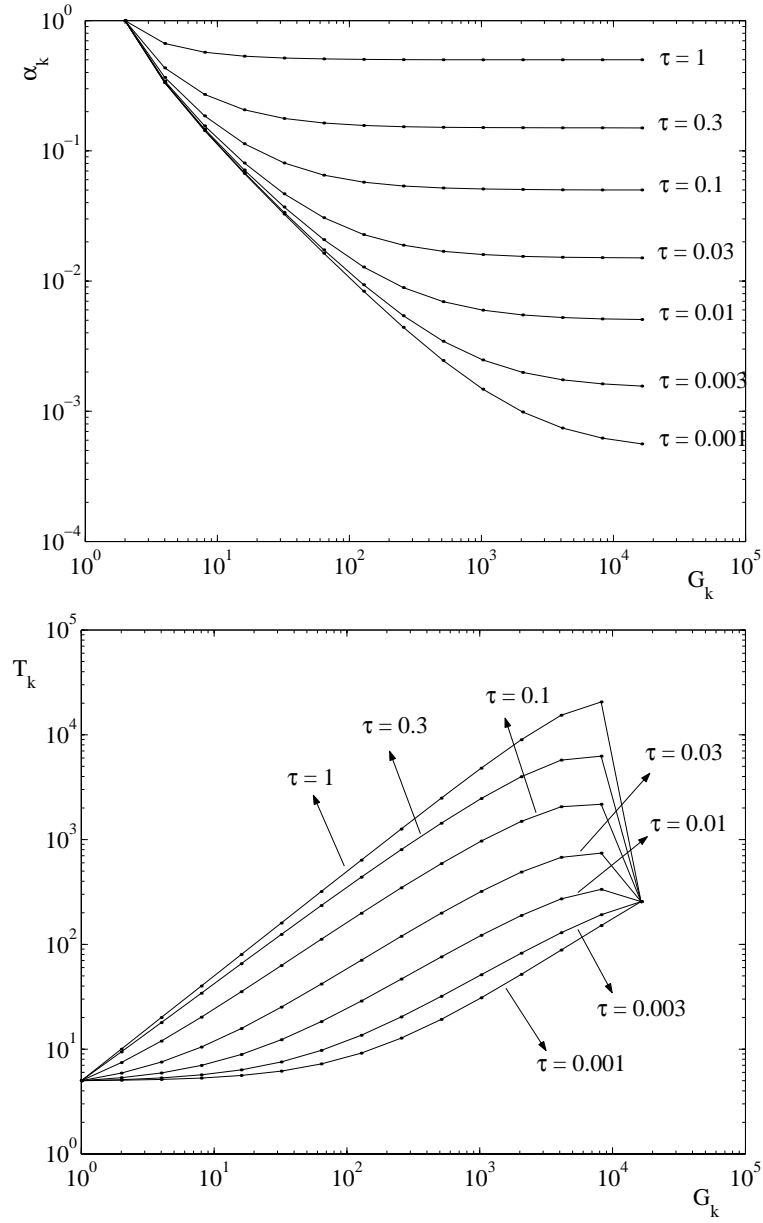


Figure 4.7: α_k (top) and corresponding T_k (bottom) resulting from α_k -model (4.19), for a range of τ -values ($G_c = 16384$, $t_G = 5$).

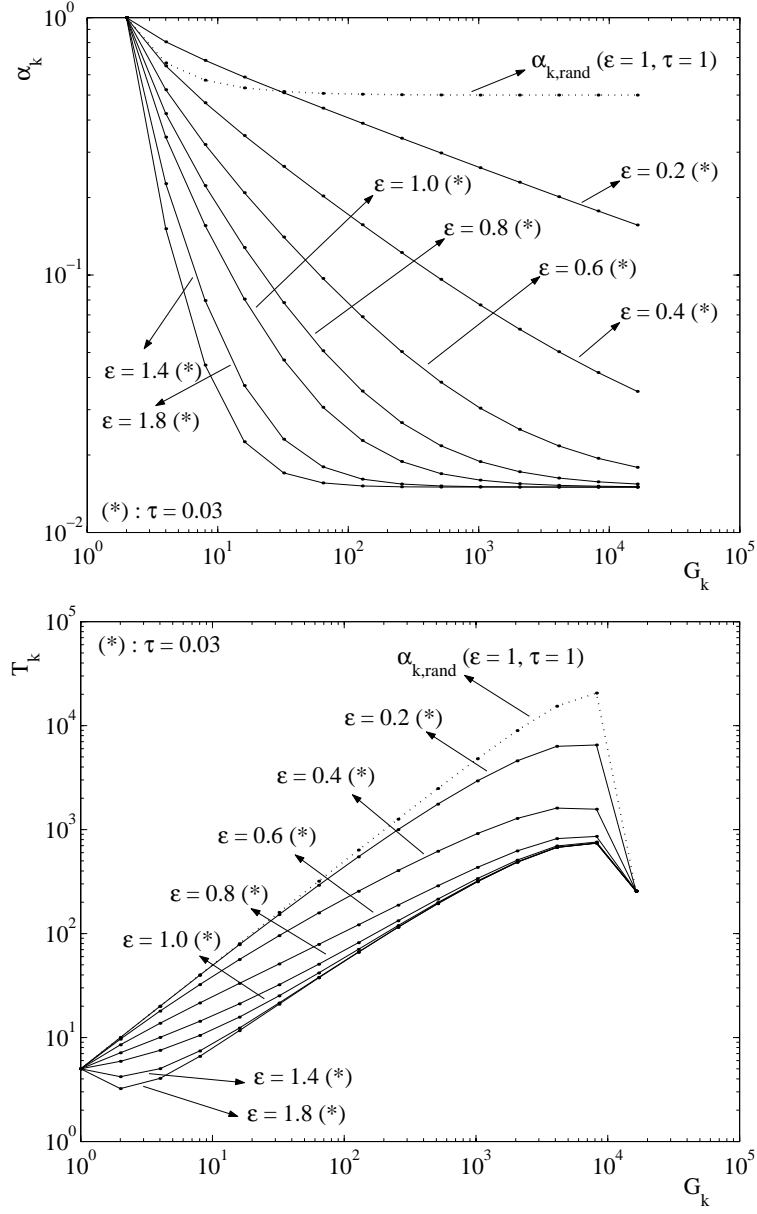


Figure 4.8: α_k (top) and corresponding T_k (bottom) resulting from α_k -model (4.20), for $\tau = 0.03$ and a range of ϵ -values ($G_c = 16384$, $t_G = 5$). The values for a random partitioning ($\tau = \epsilon = 1$) are also shown.

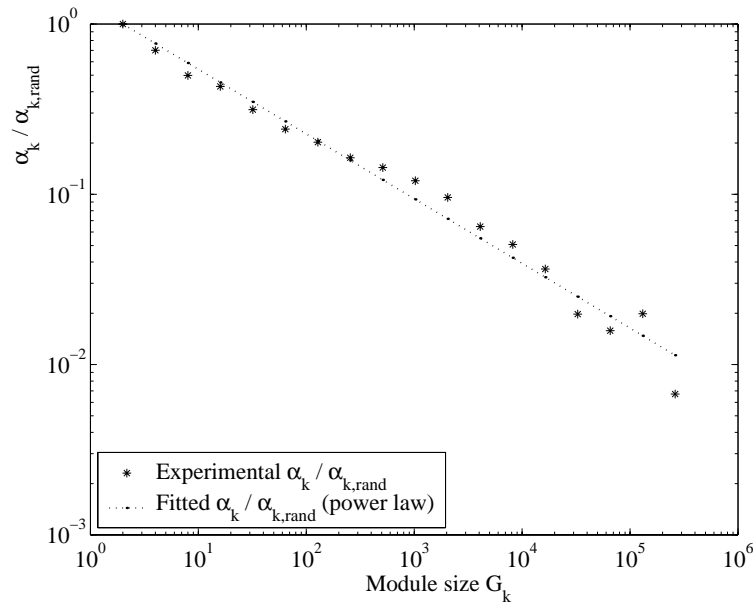


Figure 4.9: Relationship between experimental values of α_k and the corresponding values $\alpha_{k,rand}$ for random balanced bipartitioning (benchmark circuit `ibm17`, two terminal net version).

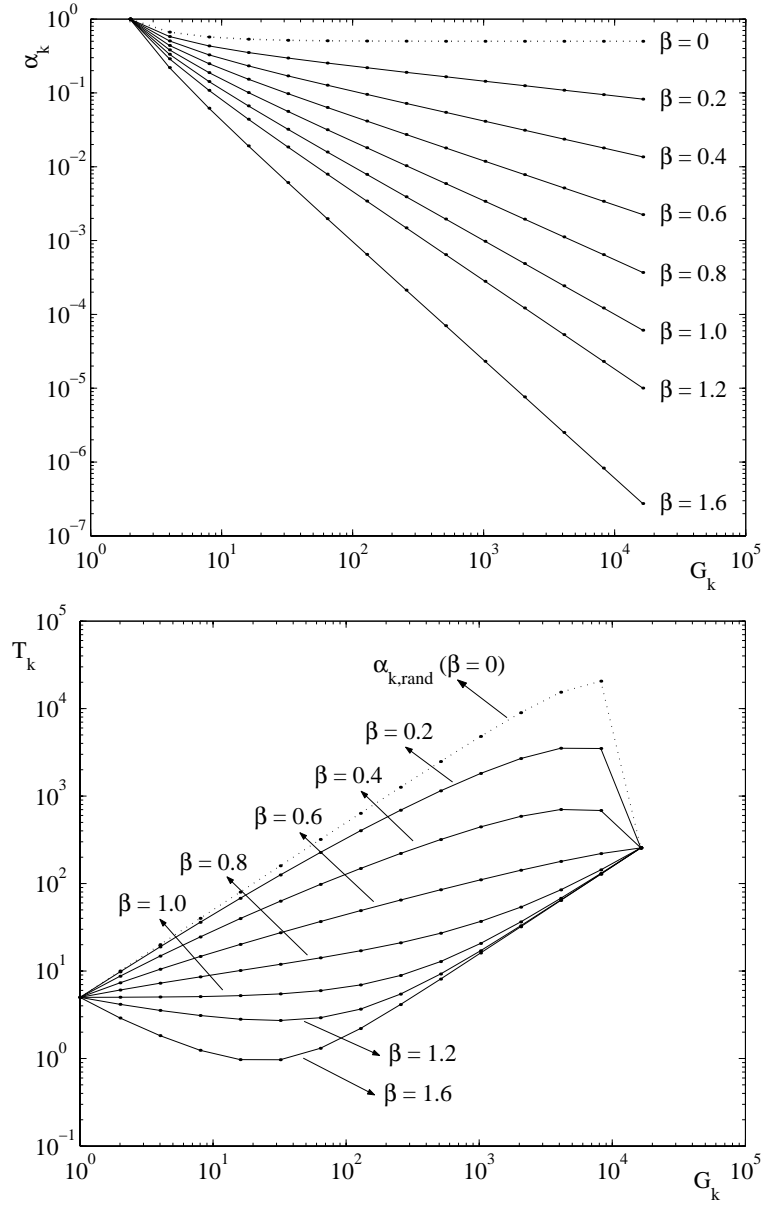


Figure 4.10: α_k (top) and corresponding T_k (bottom) resulting from α_k -model (4.23), for a range of β -values ($G_c = 16384$, $t_G = 5$). The values for a random partitioning ($\beta = 0$) are also shown.

Table 4.4: Fitted parameters and error values E_α for benchmarks ibm01 - ibm18, α_k approximated by Equations (4.20) and (4.23), using all partitioning levels.

benchmark	Verplaetse's model (4.20)			Power law model (4.23)	
	τ	ϵ	E_α	β	E_α
ibm01	4.49E-03	0.518	0.0340	0.446	0.0134
ibm02	2.91E-03	0.454	0.2039	0.388	0.1239
ibm03	3.06E-02	0.543	0.0163	0.380	0.0573
ibm04	2.07E-02	0.515	0.0832	0.389	0.0532
ibm05	6.63E-02	0.609	0.0673	0.337	0.1665
ibm06	4.57E-02	0.554	0.0659	0.354	0.0913
ibm07	1.27E-02	0.507	0.1011	0.401	0.0444
ibm08	2.13E-02	0.495	0.0577	0.368	0.0379
ibm09	3.94E-03	0.482	0.0342	0.413	0.0211
ibm10	3.94E-03	0.492	0.0865	0.422	0.0281
ibm11	5.18E-03	0.506	0.0437	0.427	0.0102
ibm12	8.63E-03	0.482	0.0701	0.393	0.0308
ibm13	4.87E-12	0.467	0.0697	0.419	0.0770
ibm14	4.53E-03	0.483	0.1364	0.408	0.0563
ibm15	4.71E-03	0.477	0.0446	0.402	0.0497
ibm16	6.12E-04	0.451	0.1406	0.400	0.0726
ibm17	2.00E-04	0.430	0.1405	0.380	0.0937
ibm18	5.51E-04	0.451	0.1466	0.401	0.0892
Average			0.0854		0.0620

Experimental validation

We have fitted both expressions (4.20) and (4.23) to the experimental α_k extracted from the Rent characteristics of benchmarks ibm01-ibm18 (two-terminal net versions). The results for benchmark ibm17, as well as the corresponding experimental and fitted Rent characteristics are also shown in figure 4.6. Note that in this figure, the fitted Rent characteristic was obtained by transforming the G_k back to the original module size domain by inverting transformation (4.17). Table 4.4 shows the fitted values of τ , ϵ and β , as well as the value of the error function E_α that was used for fitting.

It turns out that the accuracy of both models is very similar, with the power-law model (4.23), on average, slightly outperforming Verplaetse's model. However, our goal is to identify a model that yields the best estimate for the number of nets cut at each hierarchy level by using

Table 4.5: Fit results for Rent characteristics of benchmarks `ibm01` - `ibm18`, using only very few partitioning levels: error values E_α for models (4.20) and (4.23), as well as for logarithmic interpolation of the Rent characteristic.

benchmark	Verplaetse's model			Power law model			Interpolate T_k		
	levels: 2	3	4	levels: 2	3	4	levels: 2	3	4
<code>ibm01</code>	0.043	0.043	0.043	0.017	0.017	0.015	0.568	0.328	0.213
<code>ibm02</code>	0.206	0.209	0.219	0.150	0.127	0.124	1.115	0.427	0.187
<code>ibm03</code>	0.290	0.044	0.035	0.096	0.082	0.072	0.236	0.147	0.082
<code>ibm04</code>	0.162	0.187	0.195	0.054	0.057	0.061	0.652	0.266	0.121
<code>ibm05</code>	0.590	0.584	0.540	0.234	0.238	0.219	0.393	0.155	0.080
<code>ibm06</code>	0.072	0.083	0.110	0.124	0.111	0.106	0.411	0.230	0.142
<code>ibm07</code>	0.200	0.189	0.185	0.048	0.047	0.047	0.422	0.233	0.155
<code>ibm08</code>	0.203	0.203	0.179	0.046	0.046	0.040	0.546	0.345	0.257
<code>ibm09</code>	0.057	0.048	0.046	0.022	0.023	0.024	0.661	0.465	0.341
<code>ibm10</code>	0.089	0.091	0.100	0.031	0.031	0.029	0.655	0.393	0.277
<code>ibm11</code>	0.096	0.095	0.092	0.010	0.010	0.010	0.485	0.310	0.186
<code>ibm12</code>	0.072	0.077	0.075	0.033	0.031	0.031	0.550	0.370	0.295
<code>ibm13</code>	0.091	0.078	0.073	0.149	0.118	0.102	1.006	0.848	0.478
<code>ibm14</code>	0.219	0.140	0.157	0.060	0.059	0.057	0.682	0.495	0.312
<code>ibm15</code>	11.19	0.850	0.112	0.051	0.052	0.052	0.510	0.479	0.420
<code>ibm16</code>	0.142	0.143	0.141	0.103	0.104	0.087	0.955	0.686	0.427
<code>ibm17</code>	0.142	0.142	0.141	0.107	0.119	0.107	0.917	0.702	0.585
<code>ibm18</code>	0.157	0.150	0.147	0.122	0.117	0.105	1.273	0.754	0.560
Average	0.779	0.186	0.144	0.081	0.077	0.072	0.669	0.424	0.284

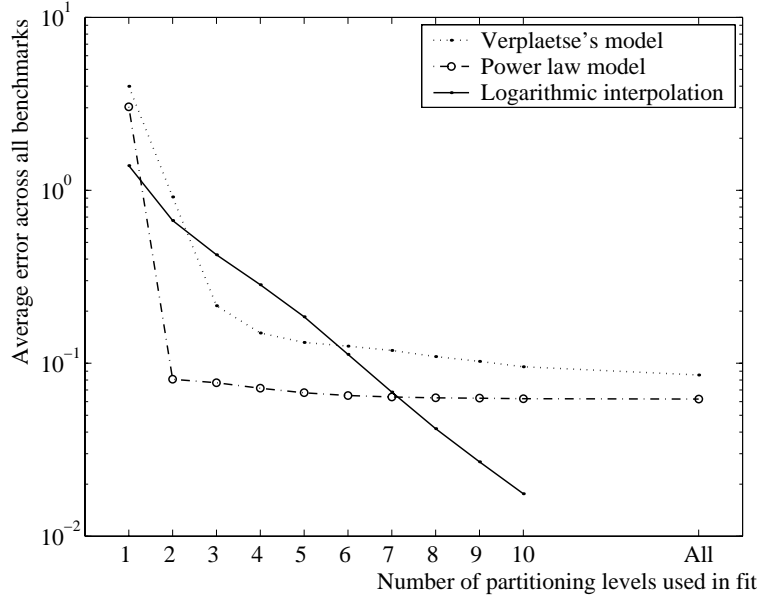


Figure 4.11: Fit results for Rent characteristics as a function of the number of partitioning levels used: E_α averaged across benchmarks `ibm01` - `ibm18` for models (4.20), (4.23), and for logarithmic interpolation of the Rent characteristic.

only a limited number of partitioning levels. Table 4.5 shows the values of E_α resulting from the α_k -models (4.20) and (4.23) when only a few partitioning levels are used. For comparison, the α_k -values were also estimated by interpolating the Rent characteristic. The convergence of these three models, as a function of the number of partitioning levels and averaged across all benchmarks, is shown in Figure 4.11.

We can conclude from Figure 4.11 and Table 4.5 that our power law model for α_k , on average, results in better estimates for very few partitioning levels. Indeed, it turns out that after three partitioning levels, the estimations do not improve much by including more levels. However, this only means that our model is more useful for obtaining rapid and accurate estimations of circuit partitioning Rent characteristics. Our model has the additional benefit of containing only a single parameter (β), compared to the two parameters in Verplaetse's model (τ and ϵ). Hence it replaces the original t and p of Rent's rule by N_c and β .

We must point out that, despite the benefits of our model over Ver-

plaetse's model, both are equally empirical. Verplaetse's ideas have, in our opinion justly, changed the focus from the terminal-gate relationship to the cut probabilities. However, there is still no thorough understanding of the fundamental mechanisms that cause these probabilities to be what they are.

Also, it would be premature in this stage to forget about the Rent exponent entirely. It has proved to be a very useful parameter for understanding the scaling behavior of, e.g., average interconnection length. Furthermore, throughout the years, the research community has come to a deeper understanding of the meaning of the Rent exponent and of its extreme values $p \simeq 0$ and $p \simeq 1$. As we will see in Section 4.3, there exist other very meaningful threshold values for this exponent. We expect to see a similar understanding of the cut probabilities and their characteristic scaling parameters emerge as the result of further research. In Chapter 7, we will discuss several results from other research domains that may be useful starting points in our search for a better understanding of (circuit) graph partitioning properties.

Though we know that Rent's rule is not an accurate model for real experimental data, it can be used to give an impression of the relationship between the Rent exponent p and the fitted model parameters τ and ϵ from model (4.20) or β from model (4.23). The cut probabilities α_k for a (hypothetical) strictly Rentian circuit are given by Equation (4.11).

We have fitted both model versions (4.20) and (4.23) to Equation (4.11) for a range of p -values, using $G_c = 16384$ and $t_G = 5$. Figure 4.12 shows the evolution of the resulting fitted parameter values as a function of p . The second parameter in Verplaetse's model, τ , only becomes important for very high values of p . As could be expected, the trends for ϵ and β are very similar. Both parameters express the fact that the cut probability decreases approximately according to a power law for sufficiently large module sizes. A similar power law scaling behavior was reported in the literature for the cut probabilities observed when bipartitioning random graphs as well as some classes of non-random graphs (e.g., [Boe01]), as a function of the graph size (see Chapter 7).

From Equation (4.23), we know that $\beta = 0$ corresponds to random partitioning or $p = 1$. The other extreme, $p = 0$ is less obvious from Figure 4.12. For a Rentian circuit with $p = 0$, the number of terminals is independent of the module size. This is typical for, e.g., a string of gates. Such a string has $T_k = 2$ and $N_k = G_k - 1$. The number of cut

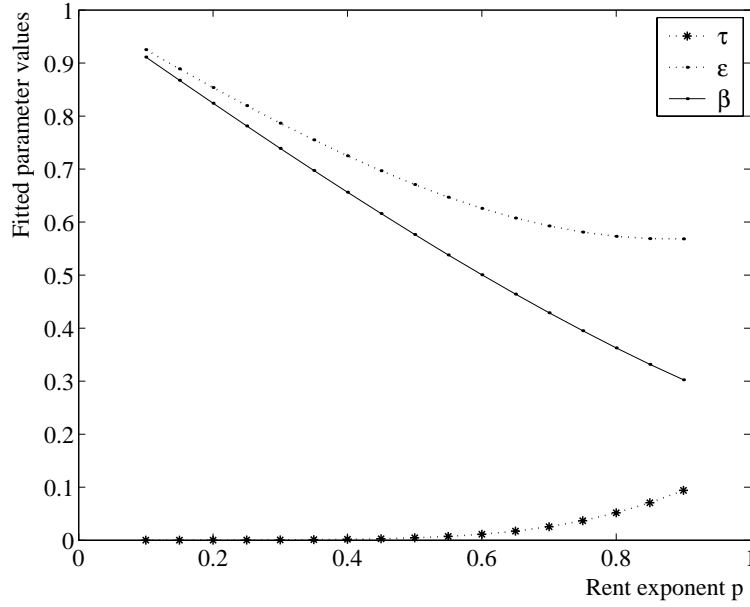


Figure 4.12: Fitted values of the new parameters τ and ϵ from model (4.20) or β from model (4.23), as a function of the Rent exponent p , for strictly Rentian circuits.

nets always equals 1, i.e.,

$$\begin{aligned} \alpha_k N_k &= 1 \\ \Leftrightarrow \alpha_k &= \frac{1}{G_k - 1}. \end{aligned} \quad (4.24)$$

Matching this expression with Equation (4.23) results in $\beta = 1$. However, the fitted β values in Figure 4.12 do not converge towards 1 as p approaches 0.

4.3 Geometric circuit modules

In this Section, we go in search of a better model for layout Rent characteristics. First, we try to establish a better parametric model for this type of terminal-gate relationship, starting from a model proposed by Christie in [Chr01]. Because this still offers no indication of how the model parameters are to be estimated *a priori*, we then proceed to a more accurate model that links the average layout Rent characteristic to the topological partitioning Rent characteristic through the wire length distribution. Based on this model, we gain some valuable insights regarding the properties of geometric Rent characteristics.

4.3.1 Christie's differential equation for layout Rent characteristics

Christie's original model

In [Chr01], Christie also indicated that the power law form of Rent's rule was probably not sufficiently accurate to obtain good predictions of the wire length distribution. In particular, he assumed that the absence of a region II might be the cause of some of the inaccuracies in Davis' model. He proposed an analytical model for (average) geometric Rent characteristics that does contain a second region. His approach was to derive a differential equation which expresses the increase of the number of module terminals as a layout region grows from an initial seed gate location to the entire placement grid. Figure 4.13 shows such a region \mathcal{G} , with region size G , which is expanded to a slightly larger one $\mathcal{G} + d\mathcal{G}$ with size $G + dG$. It is implied that $d\mathcal{G}$, the part that is added to \mathcal{G} , is adjoining to it.

To derive his equation, Christie assumed that the circuit's external terminals T_{ext} were not optimized during placement. Hence, the gates they are connected to are distributed uniformly across the entire placement grid, and the contribution of those terminals to the module terminals T is proportional to the module size. Using T' to indicate the number of module terminals that are internal terminals for the circuit⁵,

⁵This notation is in line with our earlier convention of using t'_G to denote the average number of *internal* terminals per gate. Note however, that Christie uses a different convention. In his paper, T' denotes the number of module terminals that is *external* for the circuit.

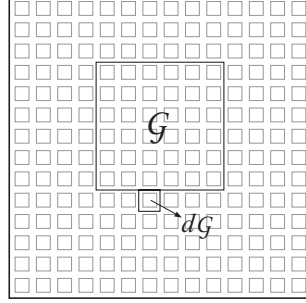


Figure 4.13: A layout region \mathcal{G} and its expansion by adding a small adjoining region $d\mathcal{G}$.

we find:

$$T = T' + \frac{G}{G_c} T_{ext} \quad (4.25)$$

Christie's differential equation for T' as a function of G is:

$$\frac{dT'}{dG} = T_{gen} + T_{term}. \quad (4.26)$$

In this expression, T_{gen} represents the number of new module terminals that are generated by increasing the region size with dG . T_{term} is the number of module terminals that disappear (is *terminated*) by expanding the region. For circuits with two-terminal nets only, T_{gen} equals the number of nets connecting the region $d\mathcal{G}$ with gates that are not in $\mathcal{G} + d\mathcal{G}$, while T_{term} is the number of nets that connects gates in \mathcal{G} to gates in $d\mathcal{G}$.

On average, the gates inside \mathcal{G} each contribute T'/G terminals to T' . Hence, a first approximation could be that $T_{gen} = T'/G$. However, in an optimized placement, the number of nets connecting $d\mathcal{G}$ to \mathcal{G} should be optimized (because $d\mathcal{G}$ is adjoining to \mathcal{G}). Therefore, to model placement optimization, Christie introduces an additional factor $\rho \in [0, 1]$:

$$T_{gen} = \rho \frac{T'}{G} \quad (4.27)$$

The term T_{term} is derived by considering the exterior of \mathcal{G} as a second region \mathcal{G}_e . This region's size is $G_e = G_c - G$. As \mathcal{G} grows, \mathcal{G}_e becomes smaller and its number of terminals decreases. The gates in $d\mathcal{G}$ originally belonged to \mathcal{G}_e but are now extracted from that region, leading to a decrease of its terminals equal to T_{term} . Now, on average,

the number of terminals each gate in \mathcal{G}_e contributes to the number of terminals of that region equals $T'/(G_c - G)$. Christie again uses the factor ρ to include placement optimization, which leads to:

$$\frac{dT'}{dG} = \rho \frac{T'}{G} - \rho \frac{T'}{G_c - G}. \quad (4.28)$$

After integration, and using the boundary condition that $T'(1) = t'_G$, this differential equation yields:

$$T' = t'_G G^\rho \left(\frac{G_c - G}{G_c - 1} \right)^\rho \quad (4.29)$$

Resulting in a total number of terminals:

$$T = t'_G \left(\frac{G(G_c - G)}{G_c - 1} \right)^\rho + \frac{G}{G_c} T_{ext} \quad (4.30)$$

This equation also satisfies the boundary condition $T(G_c) = T_{ext}$. Additionally, for $\rho = 1$, it corresponds to the average number of terminals for a layout module in a random placement. Indeed, consider a gate in such a region. The probability that a (two-terminal) net, connected to that gate, is also connected to a gate outside the region equals $(G_c - G)/(G_c - 1)$. Hence, since there are G gates in the region, the expected number of terminals T' for that region equals:

$$T' = t'_G G \frac{G_c - G}{G_c - 1}, \quad (4.31)$$

and hence:

$$T = t'_G G \frac{G_c - G}{G_c - 1} + \frac{G}{G_c} T_{ext}. \quad (4.32)$$

Note that this equation is the same as Equation (4.3), which expresses the average terminal-gate relationship for random circuit modules.

Figure 4.14 shows the average layout Rent characteristics for several synthetic benchmarks, extracted for square regions from SA-placements, as well as two fitted versions of Equation (4.30): one with only ρ as a parameter, and one where t'_G is replaced by a second fitting parameter t' . Clearly, since it includes a second region, Christie's model is a significant improvement over a simple power law. In fact, the overall shape of the measured rent characteristics is approximated quite well by Christie's model. However, when the boundary conditions at $G = 1$ and $G = G_c$ are both enforced in the fitting process (as

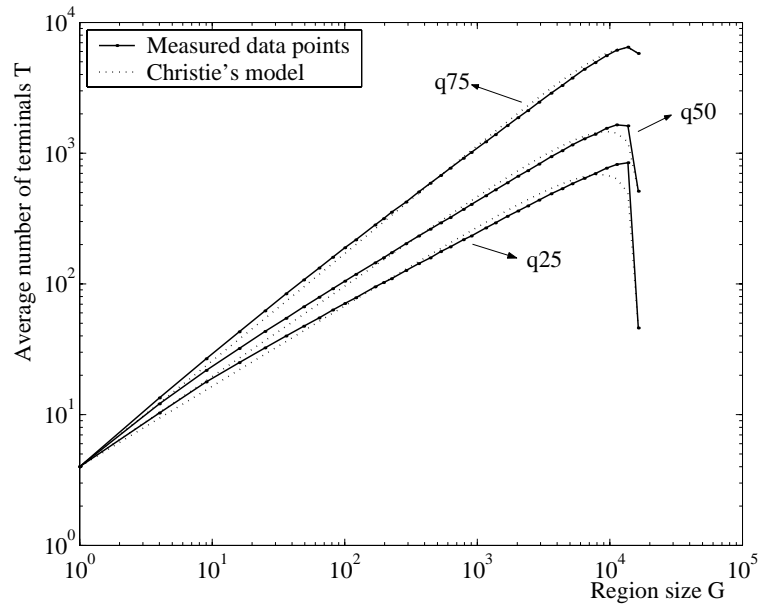


Figure 4.14: Measured average layout Rent characteristics for square regions and optimal fit of Christie's model for three synthetic benchmark circuits: q25 ($\rho = 0.615$, $E_T = 0.032$), q50 ($\rho = 0.686$, $E_T = 0.022$) and q75 ($\rho = 0.786$, $E_T = 0.012$).

was the case for the fitted curves shown in Figure 4.14), there seem to be some relatively small but systematic fitting errors. The experimental Rent characteristics show some sort of *region III*, which is not included in the model.

A recursive equation based on Christie's model

We suspect that a possible cause for the fact that Christie's model does not match reality more closely is the fact that the parameter ρ is assumed to be the same for both the generation and the termination terms. Also, it is not unlikely that ρ even depends on G . To validate these assumptions are justified, we have developed an alternative (probabilistic) version of Christie's model. Because in our architecture model, the region size is discrete, we have opted for a discrete recursive equation for the terminal-gate relationship.

Once more, we consider the situation in Figure 4.13, where a region \mathcal{G} of size G is extended by including one of its adjoining gate locations. If, on average for regions of size G , this extra gate is connected to \mathcal{G} by $T_{term}(G)$ terminals and connected to the remainder of the circuit by $T_{gen}(G)$ terminals, we find that:

$$\begin{aligned} T(G+1) &= T(G) + T_{gen}(G) - T_{term}(G) \\ &= T'(G) + T'_{gen}(G) - T'_{term}(G) + \frac{G+1}{G_c} T_{ext} \\ &= T'(G) + t'_G P_{gen}(G) - t'_G P_{term}(G) + \frac{G+1}{G_c} T_{ext}. \end{aligned} \quad (4.33)$$

Now, if only two-terminal nets are considered, $P_{gen}(G)$ expresses the *probability* that an internal net, connected to an adjoining gate of region \mathcal{G} is not connected to \mathcal{G} , while $P_{term}(G)$ is the probability that such a net is indeed connected to \mathcal{G} . Clearly,

$$P_{gen}(G) + P_{term}(G) = 1, \quad (4.34)$$

resulting in:

$$\begin{aligned} T(G+1) &= T'(G) + t'_G (1 - 2P_{term}(G)) + \frac{G+1}{G_c} T_{ext} \\ &= t'_G \left(1 + \sum_{i=1}^{G-1} (1 - 2P_{term}(i)) \right) + \frac{G+1}{G_c} T_{ext}. \end{aligned} \quad (4.35)$$

Several boundary conditions apply to $P_{term}(G)$. First, there are the boundary conditions at $G = G_c - 1$ and $G = 0$. When $G = (G_c - 1)$, only a single gate location remains outside the region. Hence, each internal net, connected to this location, must necessarily be connected to a gate inside the region, and $P_{term}(G_c - 1) = 1$. When $G = 0$, the first gate in the region is yet to be selected. Whichever gate is chosen, all nets connected to that first gate are necessarily connected a gate outside the region, so $P_{term}(0) = 0$.

Second, we know that $T(G_c) = T_{ext}$, so:

$$\begin{aligned} t'_G \left(1 + \sum_{i=1}^{G_c-1} (1 - 2P_{term}(i)) \right) &= 0 \\ \Leftrightarrow t'_G \left(1 + (G_c - 1) - 2 \sum_{i=1}^{G_c-1} P_{term}(i) \right) &= 0 \\ \Leftrightarrow \sum_{i=1}^{G_c-1} P_{term}(i) &= \frac{G_c}{2} \end{aligned} \quad (4.36)$$

We can consider two extreme situations for $P_{term}(G)$. One is that of a random placement, for which we already know that

$$P_{term,random}(G) = \frac{G}{G_c - 1}. \quad (4.37)$$

The reader can verify that this expression satisfies the boundary conditions for $G = 0$ and $G = (G_c - 1)$ as well as condition (4.36).

The other extreme is a placement for which all nets have unit length. This is the best possible wire length distribution that could appear for any circuit (and actually shows up for, e.g., a 2-D mesh or a string of gates). For such a wire length distribution, only nearest-neighbor connections are present. In this case, each gate location adjoining the region \mathcal{G} has at least one and at most three direct neighbors inside the region. The fraction of the neighbors of such adjoining gates that are inside the region, averaged across all region positions and all possible adjoining gates, yields the probability $P_{term}(G)$. We can expect this average to be close to 0.5. Hence, as an approximation for this situation, we propose:

$$P_{term,optimal}(G) = \begin{cases} 0, & G = 0, \\ 0.5, & 0 < G < G_c - 1 \\ 1, & G = G_c - 1 \end{cases} \quad (4.38)$$

This expression also satisfies all previously determined requirements.

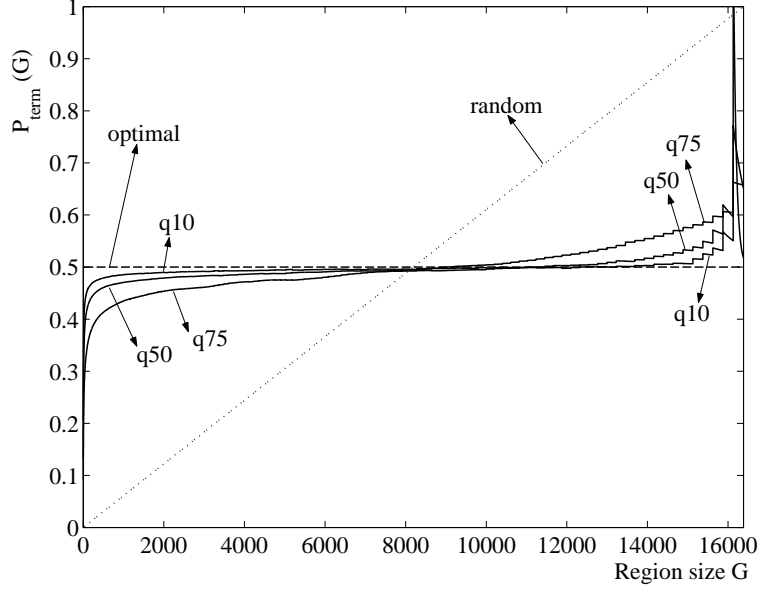


Figure 4.15: Empirical measurement of $P_{term}(G)$, obtained from average layout Rent characteristics for square regions, resulting from placements with SA. The irregularities for large regions are probably a result of the need to interpolate the Rent characteristic.

Just as we did for partitioning Rent characteristics, we can now study $P_{term}(G)$ by empirical observation. For this purpose, we have used the measured average layout Rent characteristics for square regions, resulting from placements with SA. Because we now require a terminal-gate relationship for *all* region sizes, the missing data points were interpolated in the logarithmic domain. Since we are interested in finding an expression for $P_{term}(G)$ that links the geometric Rent characteristic to the circuit topology, we have measured $P_{term}(G)$ for the synthetic benchmark circuits q10 - q90. The extracted terminal-gate relationships for some of these benchmarks are shown in Figure 4.15. The relationship between P_{term} and the region size G is similar in all cases. For the synthetic benchmark circuits with very low (partitioning) Rent exponents $p_{(p)}$, $P_{term}(G)$ is very close to $P_{term,optimal}(G)$. Although it evolves away from this approximated optimal relationship as the partitioning Rent exponent increases, it remains far from the worst case behavior $P_{term,random}(G)$.

Unfortunately, in contrast to the cut probabilities in Verplaetse's re-

cursive partitioning model, there does not seem to be a straightforward mathematical expression for $P_{term}(G)$, as a function of, for instance, the circuit's partitioning Rent exponent. Furthermore, like Davis' technique, neither Christie's model nor our discrete version of it give any mathematical indication of the impact of region shape or the placement quality on $P_{term}(G)$ or the geometric Rent characteristic. To gain a better understanding of how these factors, as well as the circuit topology are reflected in a geometric Rent characteristic, we develop a more complete model in the next section.

4.3.2 Enumeration model for layout Rent characteristics

In this section, we first derive a model for the expected number of terminals for a single geometric module, as a function of its shape and its position in the placement grid. Our approach, based on the enumeration of terminals, is very similar to Davis' concept of terminal conservation. However, we consider the layout Rent characteristics to be a consequence of the wire length distribution, rather than its cause. Hence, in contrast to Davis, we use the wire length distribution to derive the terminal-gate relationship.

We then derive an approximation for geometric (layout partitioning or average layout) Rent characteristics. This is done by taking the (approximated) average of the terminal model across all possible region positions for a given module generation strategy (either layout partitioning modules or all possible modules).

In contrast to Christie's model, or our extension of it, this new theoretical model is not meant to be used for the a priori prediction of interconnect properties. Instead, its purpose is to gain more insight in the properties of average layout Rent characteristics obtained from circuit placements.

Terminology

Consider the set of all regions in a circuit layout that have size G and equal shape, but with different relative positions within the placement grid. For each of these regions \mathcal{G}_i ($i = 1 \dots I$), a module can be created that consists of all gates within that region.

Let g_k ($k = 1 \dots G$) be the relative gate locations inside the region. Because the regions \mathcal{G}_i all have the same shape and size, these relative

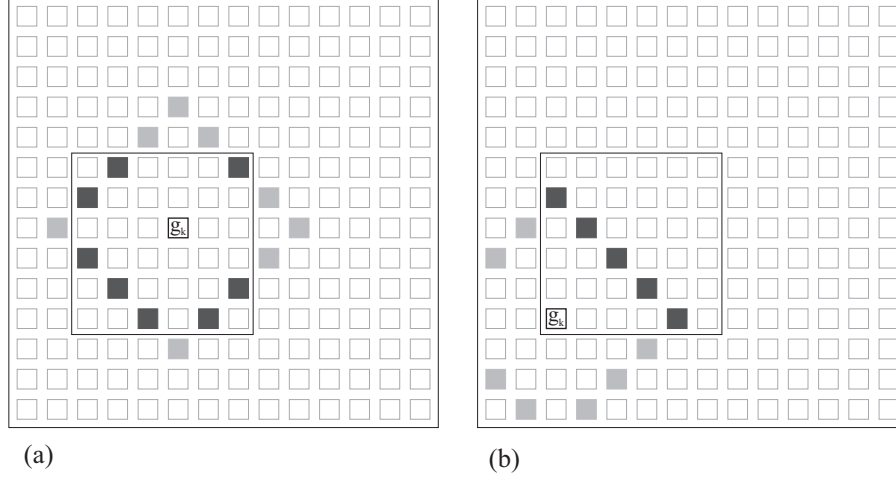


Figure 4.16: Illustration of a layout region and the calculation of $P_{k,i}\{int|length = \ell\}$, Equation (4.42), for $\ell = 4$ and for two different gate locations in the region. Gate locations that are counted in $\Phi_k(\ell = 4)$ are shown in black while those counted in $\Phi_{k,i}(\ell = 4)$ are gray or black. (a): $\Phi_{k,i}(\ell = 4) = 16$, $\Phi_k(\ell = 4) = 8$ and $P_{k,i}\{int|length = 4\} = 0.5$; (b): $\Phi_{k,i}(\ell = 4) = 12$, $\Phi_k(\ell = 4) = 5$ and $P_{k,i}\{int|length = 4\} = 0.417$.

gate locations are the same for all regions in the set. For each g_k , we can now define $\Phi_k(\ell)$ to be the local site function⁶ of gate location g_k with respect to the region. Hence, it gives the number of gates *in the region* that are a distance ℓ apart from gate g_k and is independent of the region position i . Similarly, $\Phi_{k,i}(\ell)$ is the local site function of gate location g_k in region \mathcal{G}_i , *with respect to the entire placement grid*. It gives the number of gates *in the entire grid* that are a distance ℓ apart from gate location g_k . This second site function *does* depend on the region position i . Figure 4.16 shows an example for a square region, but our model is applicable to regions of any shape or size.

Terminals for a single module

For a given region \mathcal{G}_i , let $T_{int}(i)$ be the number of gate terminals that is used for connections between gates inside the region (*internal terminals*). Since only two-terminal nets are considered, the number of internal nets for the corresponding module is given by $T_{int}(i)/2$. The

⁶Remember that a local site function is a site function that only includes net sites connected to one specific gate location.

relation between $T(G, i)$, the number of module terminals, and $T_{int}(i)$ is given by

$$\begin{aligned} T(G, i) &= t_G G - T_{int}(i) \\ &= t'_G G - T_{int}(i) + \frac{G}{G_c} T_{ext}, \end{aligned} \quad (4.39)$$

where t'_G denotes the average number of terminals per gate that are internal to the circuit, given by Equation (2.1).

$T_{int}(i)$ can be calculated as the sum of the internal terminals $T_{int}(k, i)$ for each gate location g_k in the region:

$$\begin{aligned} T_{int}(i) &= \sum_{g_k \in \mathcal{G}} T_{int}(k, i) \\ &= \sum_{g_k \in \mathcal{G}} t'_G P_{k,i} \{int\}. \end{aligned} \quad (4.40)$$

The probability $P_{k,i} \{int\}$ must be interpreted as the relative frequency, over many different placement experiments, that a globally internal wire connected to the gate in grid location g_k of region \mathcal{G}_i , stays inside \mathcal{G}_i .

The expression for $T_{int}(k, i)$ can be further refined as follows:

$$\begin{aligned} T_{int}(k, i) &= t'_G \sum_{\ell=1}^{\ell_{max}} P_{k,i} \{length = \ell, int\} \\ &= t'_G \sum_{\ell=1}^{\ell_{max}} P_k \{length = \ell\} P_{k,i} \{int | length = \ell\} \end{aligned} \quad (4.41)$$

In this equation, $P_{k,i} \{length = \ell\}$ is the *local wire length distribution* $\mathcal{N}_{k,i}(\ell)$ at gate location g_k of region \mathcal{G}_i . This is the probability that a wire connected to a gate in this location has length ℓ . $P_{k,i} \{int | length = \ell\}$ is the conditional probability that a wire of length ℓ , connected to a gate in location g_k , is connected to another gate inside the region. For each length ℓ , this probability is given by the fraction of the gate locations at a distance ℓ from g_k that are internal to the region. To obtain it, we need to calculate the two local site functions $\Phi_k(\ell)$ and $\Phi_{k,i}(\ell)$. For each length, the proportion between both site functions yields the desired fraction:

$$P_{k,i} \{int | length = \ell\} = \frac{\Phi_k(\ell)}{\Phi_{k,i}(\ell)} \quad (4.42)$$

An illustration of this is shown in Figure 4.16 for two different gate locations g_k and for $\ell = 4$. For the first gate location (configuration (a) in this Figure), we find that $P_{k,i}\{\text{int}|\text{length} = 4\} = 0.5$. For the second gate location (in the same region), shown in configuration (b) in Figure 4.16, $P_{k,i}\{\text{int}|\text{length} = 4\} = 0.417$.

We can now calculate $T_{\text{int}}(i)$:

$$T_{\text{int}}(i) = t'_G \sum_{g_k \in \mathcal{G}_i} \sum_{\ell=1}^{\ell_{\max}} \mathcal{N}_{k,i}(\ell) \frac{\Phi_k(\ell)}{\Phi_{k,i}(\ell)}. \quad (4.43)$$

Finally, combining Equations (4.43) and (4.39), we find the final expression for $T(G, i)$:

$$T(G, i) = t'_G \left(G - \sum_{g_k \in \mathcal{G}_i} \sum_{\ell=1}^{\ell_{\max}} \mathcal{N}_{k,i}(\ell) \frac{\Phi_k(\ell)}{\Phi_{k,i}(\ell)} \right) + \frac{G}{G_c} T_{\text{ext}} \quad (4.44)$$

For rectangular regions, this expression is relatively easy to calculate, using the techniques described in Appendix C. However, note that this is not restrictive: our model can be applied to regions of any given shape, possibly at the cost of a much larger computation time.

The result of Equation (4.44) obviously depends on the shape and size of region \mathcal{G}_i , as well as on its relative position within the placement grid.

Average across all modules considered

A layout Rent characteristic is an average relationship between the number of terminals of a layout region and the number of gates in that region. For an average layout Rent characteristic, the average must be taken across *all* possible positions i of a given region shape in the placement grid. Remember that in Equation (4.44), the labels g_k reflect the relative position of a gate location *within the region*. Because we now consider many different region positions, the symbol g_k no longer unambiguously points to a single grid location in the placement grid. The average number of terminals $T(G)$ across all regions \mathcal{G}_i is given by:

$$\begin{aligned} T(G) &= E_i [T(G, i)] \\ &= t'_G \left(G - E_i \left[\sum_{g_k \in \mathcal{G}_i} \sum_{\ell=1}^{\ell_{\max}} \mathcal{N}_{k,i}(\ell) \frac{\Phi_k(\ell)}{\Phi_{k,i}(\ell)} \right] \right) + \frac{G}{G_c} T_{\text{ext}}, \end{aligned} \quad (4.45)$$

where the average is across all positions i . Since $\Phi_k(\ell)$ is independent of the region position i , we can rewrite Equation (4.45) as follows:

$$T(G) = t'_G \left(G - \sum_{\ell=1}^{\ell_{max}} \sum_{g_k \in \mathcal{G}_i} \Phi_k(\ell) E_i \left[\frac{\mathcal{N}_{k,i}(\ell)}{\Phi_{k,i}(\ell)} \right] \right) + \frac{G}{G_c} T_{ext} \quad (4.46)$$

The remaining average

$$E \left[\frac{\mathcal{N}_{k,i}(\ell)}{\Phi_{k,i}(\ell)} \right] \quad (4.47)$$

is for a given relative gate position g_k within the region, across all region positions i in the placement grid. It captures the impact of the different region positions used in averaging.

In the previous chapter, we discussed the concept of *occupation probability* $\mathcal{Q}(\ell)$ as the fraction of the total number of sites of length ℓ that is occupied after placement. A similar concept can be defined with respect to the local site distribution. Since the average number of internal wires connected to a single gate equals t'_G , the local wire length distribution $\mathcal{N}_{k,i}(\ell)$ and the local site function $\Phi_{k,i}(\ell)$ are related by the following expression:

$$t'_G \mathcal{N}_{k,i}(\ell) = \mathcal{Q}_{k,i}(\ell) \Phi_{k,i}(\ell). \quad (4.48)$$

Expression (4.47) can therefore be rewritten as:

$$\frac{1}{t'_G} E[\mathcal{Q}_{k,i}(\ell)], \quad (4.49)$$

representing the average local occupation probability for gate location g_k within the region, averaged across all region positions. Unfortunately, the exact calculation of Expressions (4.47) or (4.49) depends on the availability of the local wire length distributions or occupation probabilities. These are not easy to obtain experimentally, because in each individual placement experiment, only very few wires are connected to each gate. As a result, to achieve a relatively good statistical estimation of local wire length distributions, averages across a large number of placement experiments would have to be taken.

Because of this, we will approximate Expression (4.47), based on the assumption that the placement is *homogeneous*. But what does this mean mathematically? A possible assumption might be that the occupation probability for each length ℓ is independent of the location of a

site in the placement grid. However, if this were the case, the average local occupation probability in Expression (4.49) would be equal to the global occupation probability $\mathcal{Q}(\ell)$ and the resulting layout Rent characteristics would not depend in any way on the positions used in averaging. In [VDSVC01, DVSVC03], however, it was shown that there is a difference between average layout Rent characteristics and Rent characteristics obtained by recursively partitioning the layout grid (*layout partitioning Rent characteristics*). Since the region shapes can be equal in both cases, this difference must be caused by the different sets of region positions used in averaging. As a consequence, the local occupation probabilities must be position-dependent.

Alternatively, we might assume that the local wire length distribution is independent of the position $g_{k,i}$:

$$\mathcal{N}_{k,i}(\ell) = \mathcal{N}(\ell). \quad (4.50)$$

Figure 4.17 illustrates how this affects the contribution of one gate location g_k to the number of module terminals for two different region positions. It shows that, under this assumption, layout modules close to the boundary of the placement grid tend to have fewer pins than those that are closer to the center of the grid. This effect was first discussed by Verplaetse in [VDSVC01] and referred to as the *boundary effect*. We suspect that Equation (4.50) represents an oversimplification of reality, but as we will show, the resulting model results do accord with numbers of module terminals observed in real placements.

Using the assumption of a position-independent local wire length distribution, we can approximate expression (4.47) as follows:

$$E_i \left[\frac{\mathcal{N}_{k,i}(\ell)}{\Phi_{k,i}(\ell)} \right] \simeq \frac{E_i[\mathcal{N}_{k,i}(\ell)]}{E_i[\Phi_{k,i}(\ell)]} \simeq \frac{\mathcal{N}(\ell)}{E_i[\Phi_{k,i}(\ell)]}, \quad (4.51)$$

where we have replaced the average local wire length distributions by the global wire length distribution. The experimental results that are given further in this chapter indicate that the variance of $\Phi_{k,i}(\ell)$ caused by its dependency on the region position i is sufficiently small to justify this approximation.

Experiments have indicated that, to further alleviate the computational complexity of our model, the average local site function in the denominator of Expression (4.51) can be replaced by a global average $\Phi_{av}(\ell)$, where a second averaging is done across all gate locations in the

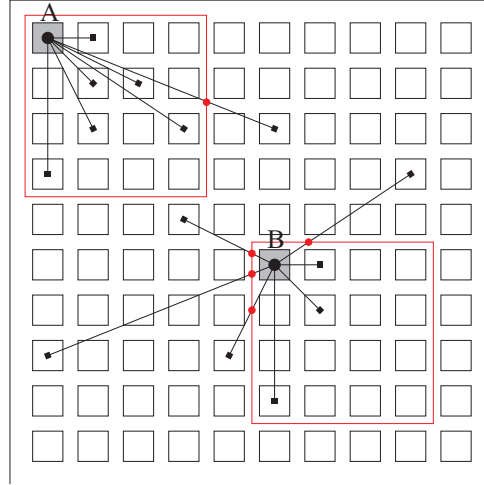


Figure 4.17: Illustration of the boundary effect, modeled by the assumption of a constant local wire length distribution.

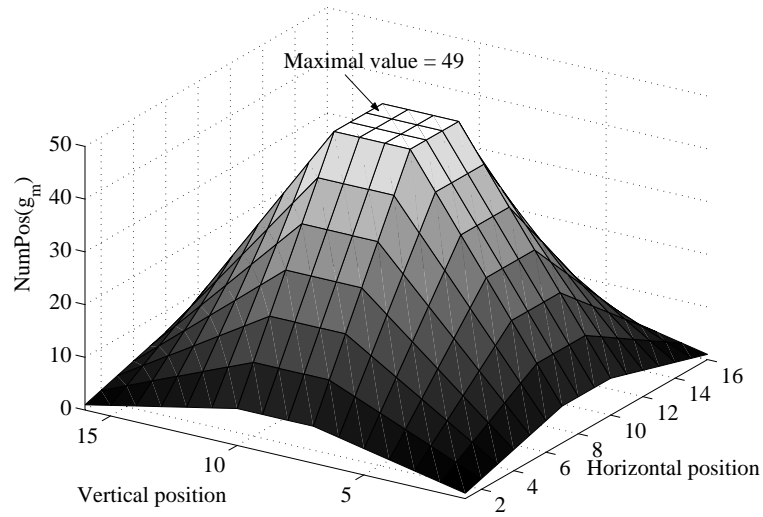


Figure 4.18: Graphical illustration of $NumPos(m)$ in the calculation of the average layout Rent characteristic for a placement grid of 16×16 grid locations. The plot shows the values of $NumPos(m)$ for square regions of 7×7 gate locations.

module:

$$\Phi_{av}(\ell) = \frac{1}{I} \sum_{i=1}^I \frac{1}{G} \sum_{g_k \in \mathcal{G}_i} \Phi_{k,i}(\ell), \quad (4.52)$$

where I denotes the total number of region positions used in averaging.

This double summation can be rearranged by considering the fact that each absolute gate location in the entire *placement grid* is visited a number of times. Consider the transformation \mathcal{R} :

$$\mathcal{R} : (k, i) \rightarrow m, \quad (4.53)$$

that maps the relative “coordinates” (k, i) of gate location g_k in region \mathcal{G}_i onto the corresponding absolute location g_m in the placement grid. The number of times the gate location g_m is visited in the double summation of Equation 4.52 is now given by:

$$NumPos(m) = |\{(k, i) : \mathcal{R}(k, i) = m\}| = |\mathcal{R}^{-1}(m)| \quad (4.54)$$

An example of $NumPos(m)$ for calculating the average layout Rent characteristic is shown in Figure 4.18. The corner locations of the placement are inside the region for a single region position only, so for these locations $NumPos(m) = 1$. It is maximal for centralized locations, which can be inside at most G different region positions. Using this function, Equation (4.52) becomes:

$$\Phi_{av}(\ell) = \frac{1}{GI} \sum_{m=1}^{S_g} NumPos(m) \Phi_m(\ell), \quad (4.55)$$

which still succeeds in capturing the impact of module selection on the Rent characteristic. What we have lost by Approximation (4.51) is any correlation there might be between the location dependency of $\mathcal{N}_{k,i}(\ell)$ and $\Phi_{k,i}(\ell)$. The introduction of $\Phi_{av}(\ell)$ disregards the correlation between $\Phi_k(\ell)$ and the average local occupation probability for a gate location g_k within the module. With these approximations, our final model expression is:

$$T(G) = t'_G \left(G - \sum_{\ell=1}^{\ell_{max}} \frac{\mathcal{N}(\ell)}{\Phi_{av}(\ell)} \sum_{g_k \in \mathcal{G}} \Phi_k(\ell) \right) + \frac{G}{G_c} T_{ext}. \quad (4.56)$$

The summation across gate locations g_k is now for a “generic” region \mathcal{G} of the considered size and shape, independent of its position.

Model validation

We have validated our model, using wire length distributions obtained from SA-placements. Figure 4.19 shows the average layout Rent characteristic for benchmark `ibm10`, placed in a square placement grid of 115×115 grid locations, and using square regions. Amongst other things, Figure 4.20 shows average layout Rent characteristics obtained for square regions for some synthetic benchmarks with different partitioning Rent exponents (placed in square grids of 128×128). In both Figures, the corresponding Rent characteristics obtained from our model are also shown. We will again consider the model error E_T across I_{max} measured data points (region sizes):

$$E_T = \frac{\sum_{\zeta=1}^{I_{max}} [\log_2(T_\zeta) - \log_2(T_{\zeta,model})]^2}{I_{max}}. \quad (4.57)$$

Table 4.6 shows these errors for the average layout Rent characteristics for the ISPD98 benchmarks, extracted for square regions. These results show that, despite the approximations that were made, the results from our model are very close to the experimental Rent characteristics.

Since the wire length distribution is used in our model, it can be expected that the resulting layout Rent characteristics depend on the placement quality. To demonstrate this impact, we have calculated the average layout Rent characteristic for benchmark `ibm01` with three different wire length distributions, obtained from real placement results. Two of them were obtained by performing *flat* SA placements, but with different cooling factors (see Appendix A): the first with the same slow cooling we have used in all other SA-experiments (cooling factor = 0.999), and the second with much faster cooling (cooling factor = 0.98). The third wire length distribution was extracted from a placement obtained with `Plato`. Figure 4.21 shows the average layout Rent characteristics for square modules, resulting from each of these three wire length distributions. Also shown are the region I Rent exponents, extracted from these Rent characteristics. Table 4.7 shows that the average layout Rent exponent correlates well with the placement quality, expressed by the average wire length.

Now that we have ascertained the accuracy of our model, represented by Equation (4.56), for average layout Rent characteristics, we will restrict all further experimental data in this section to results obtained from our model.

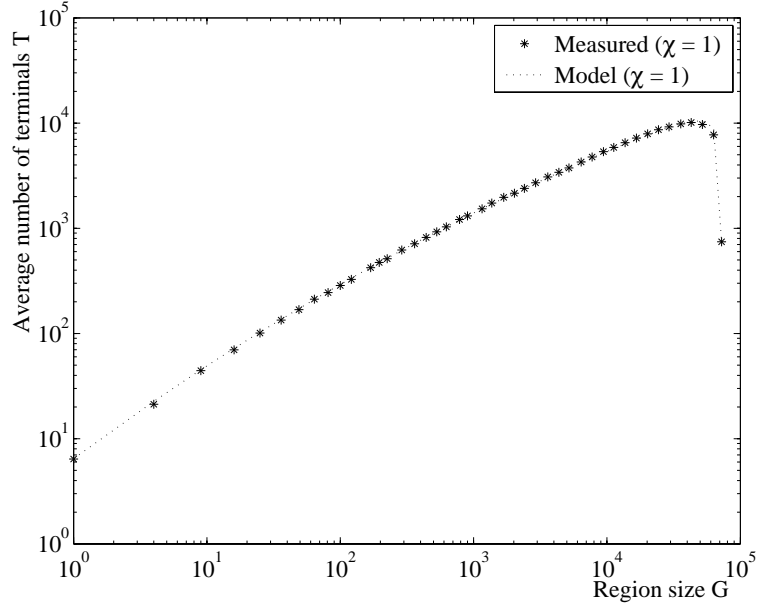


Figure 4.19: Average layout Rent characteristic for benchmark *ibm10* for placement with SA and square regions: measured data points and model results (using the measured SA wire length distribution).

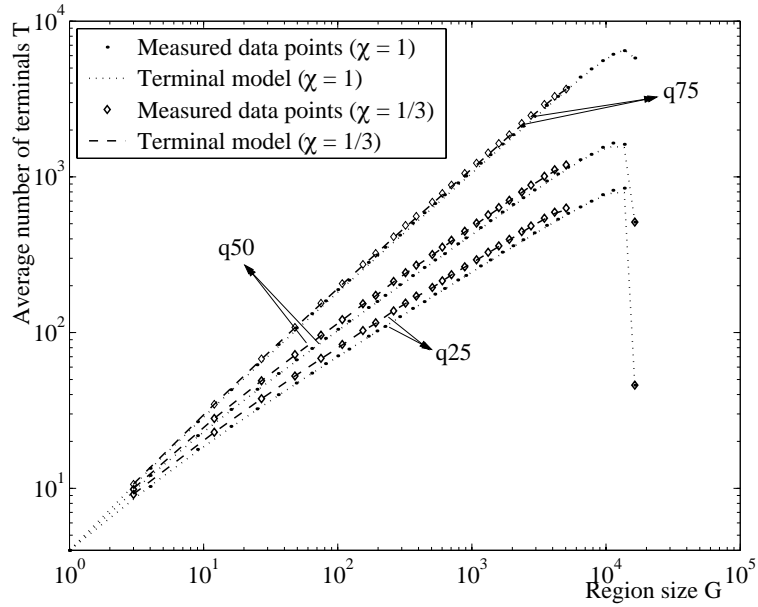


Figure 4.20: Average layout Rent characteristics for SA placements of benchmarks *q25*, *q50* and *q75*: measured data points and model results for square regions and for regions with aspect ratio $1/3$.

Table 4.6: Errors E_T on the average layout Rent characteristics for square regions and SA placements of the ISPD98 benchmarks.

benchmark	Error E_T
ibm01	0.00429
ibm02	0.0364
ibm03	0.0258
ibm04	0.00511
ibm05	0.0225
ibm06	0.0269
ibm07	0.0128
ibm08	0.0347
ibm09	0.0129
ibm10	0.00230
ibm11	0.00814
ibm12	0.0106
ibm13	0.00311
ibm14	0.00589
ibm15	0.0335
ibm16	0.00887
ibm17	0.00307
ibm18	0.00213

Table 4.7: Relationship between the average layout Rent exponent p_{al} (for square modules) and the placement quality for benchmark circuit ibm01.

Placement tool	Average wire length	Fitted p_{al}
SA (slow cooling)	5.322	0.646
SA (fast cooling)	6.232	0.664
Plato	7.032	0.682

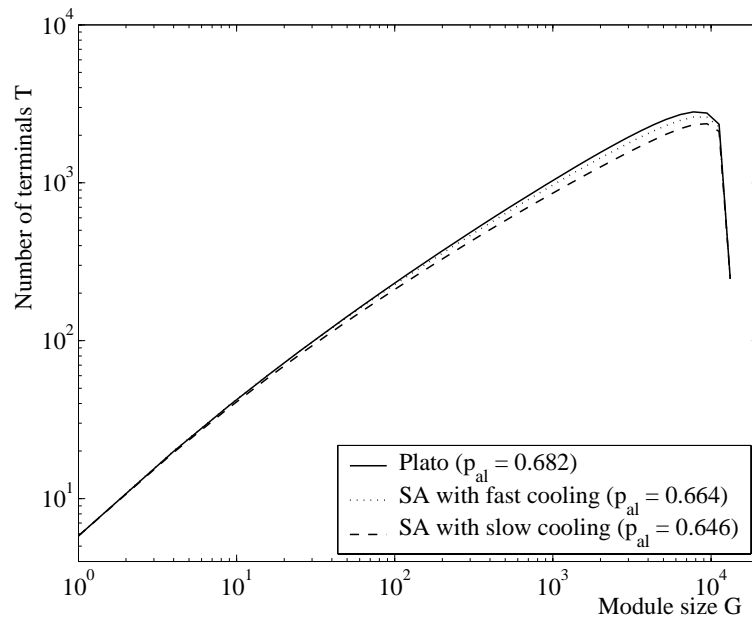


Figure 4.21: Average layout Rent characteristics (model results) for benchmark circuit *ibm01*, using measured wire length distributions from three different placement tools: SA with slow cooling, SA with fast cooling and Plato.

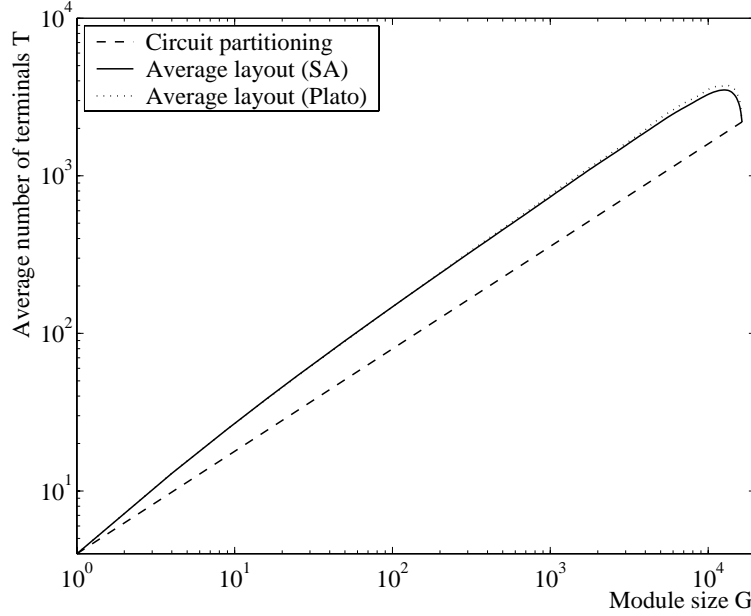


Figure 4.22: Average layout Rent characteristics for benchmark circuit `q65`, extracted from placements with `SA` and `Plato`. The partitioning Rent characteristic is also shown for comparison.

4.3.3 Properties of average layout Rent characteristics

One of the main reasons for deriving our model was to gain more insight in the properties of terminal-gate relationships of layout regions and their usefulness for predicting wire length distributions. When grouped according to size, the regions and region positions used in Davis' approach are close to random. However, they are not rectangular. Still, we will first study the properties of average layout Rent characteristics resulting from rectangular layout regions. Then, we proceed to investigating the impact of the region shape on these characteristics and finally, we apply the conclusions to the regions applied in Davis' technique.

Region I scaling behavior

Figure 4.22 compares the partitioning and average layout Rent characteristics for benchmarks `q65`. For the time being, we restrict our discussion to modules originating from square layout regions.

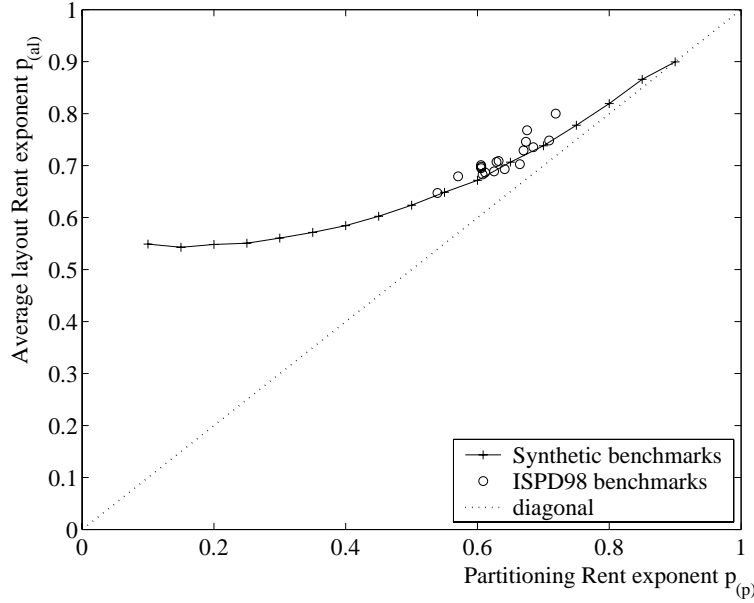


Figure 4.23: Average layout Rent exponent as a function of partitioning Rent exponent for SA placement algorithm.

Remember that for a partitioning Rent characteristic, modules are generated in such a way that their number of terminals is minimized. Hence, we can expect that the resulting number of terminals is a lower bound on average terminal numbers resulting from other module generation strategies. Figure 4.22 confirms this: the average number of terminals for layout modules is significantly higher than that for modules obtained by recursive partitioning.

Also in this Figure, the region I Rent exponent appears to be higher for the average layout Rent characteristic. We have fitted Rent's rule to the average layout Rent characteristics (for square modules), obtained from our model, for benchmarks q_{10} - q_{90} . This results in the empirical relationship between the average layout Rent exponent $p_{(al)}$ and the partitioning Rent exponent $p_{(p)}$ shown in Figure 4.23. Results for the *ispd98*-benchmarks are also shown in this Figure.

First of all, when comparing the experimental and model average layout Rent parameters, we should keep in mind that the reduction of an entire Rent characteristic to a two-parameter power law is an extremely rough approximation. The result of a fitting procedure using a limited set of data points is very sensitive to the data values. Our au-

tomated fitting procedure (see Appendix A) automatically selects the data points belonging to region I and was tuned to be as robust as possible. Still, in some cases, the inclusion or exclusion of a single data point can have a significant effect on the resulting Rent parameter values. This is the cause of the observed irregularities in the relationship between $p_{(al)}$ and $p_{(p)}$.

However, despite these irregularities, two observations can be made from the results shown in Figure 4.23. First, for very low values of the partitioning Rent exponent, $p_{(al)}$ appears to be bounded from below by a value somewhat higher than 0.5. Second, $p_{(al)}$ is generally higher than $p_{(p)}$, and converges towards it as $p_{(p)}$ approaches 1.

In contrast to what has long been assumed, the observed lower bound on the average layout Rent exponent is not merely the result of bad placement optimization. In fact, there exists a theoretical limit to the scaling behavior of the average layout Rent characteristic (i.e., the Rent exponent), caused by the embedding of a circuit in a two-dimensional layout medium. Consider a square region in a circuit layout. In a good placement, any two gates that are connected are preferably placed close together, such that the connections between them are short. We know from the previous chapter that the wire length distribution scales approximately according to the power law:

$$\mathcal{N}(\ell) \sim \ell^{2p-3}, \quad (4.58)$$

with p equal to the partitioning Rent exponent $p_{(p)}$. For placements in a D -dimensional placement grid, Stroobandt found the scaling exponent of the wire length distribution to be $Dp - (D + 1)$ [Str01b]. As a consequence of this exponential scaling law, almost all wires must be very short or even of unit length for very low values of the partitioning Rent exponent. Hence, in this case, most external terminals for the layout region must belong to gates on or very close to the boundary of that region, and the number of terminals is approximately proportional to the number of gates on the region boundary. For a square region with side w , this number is proportional to w . With the number of gates G in the region equal to w^2 , the number of terminals T is proportional to $G^{1/2}$, leading to a Rent exponent of 0.5. This restriction on the layout Rent exponent was also found by Feuer in 1982 [Feu82] for diamond-shaped regions. In [DVSVC03], this argument is generalized to placements in a D -dimensional layout medium, for which T is proportional to $G^{(D-1)/D}$.

This lower bound can also be derived from our theoretical model, by assuming the best possible wire length distribution that could appear for any circuit (and actually shows up for, e.g., a 2-D mesh or a string of gates):

$$\mathcal{N}(\ell) = \begin{cases} 1, & \ell = 1 \\ 0, & \ell > 1 \end{cases} \quad (4.59)$$

It is clear that this wire length distribution also minimizes the number of terminals emerging from any layout region. Using Equation (4.59), Equation (4.56) reduces to

$$T = t'_G \left(G - \frac{1}{\Phi_{av}(1)} \sum_{k=1}^G \Phi_k(1) \right) + \frac{G}{G_c} T_{ext}. \quad (4.60)$$

Now, external nets can originate *only* from gate locations next to the region boundary. Note that $\Phi_{av}(1)$ equals the average number of direct neighbors a gate location has, where the average is taken across all region positions and across all gate positions within those regions. Only the gate locations on the boundary of the placement grid do not have four direct neighbors. Since for all but the largest region sizes, most region positions do not intersect with the medium boundaries, we can approximate $\Phi_{av}(1) \simeq 4$. The summation in Equation (4.60) yields twice the total number of neighboring gate locations in the region. For *square regions*, we find:

$$\begin{aligned} \sum_{k=1}^G \Phi_k(1) &= 2 \times 2 (\sqrt{G} - 1) \sqrt{G} \\ &= 4 (G - \sqrt{G}) \end{aligned} \quad (4.61)$$

In this expression, $(\sqrt{G} - 1)$ is the number of adjoining gate location pairs in one dimension. For instance, it could be the number of horizontally adjoining gate location pairs in a single row of the placement grid. This number must be multiplied by the number of rows \sqrt{G} to find all horizontally adjoining pairs in the entire placement grid. Finally, this must be repeated for vertically adjoining pairs, hence the additional multiplication by two.⁷

⁷For D -dimensional layout regions, this equation yields

$$\sum_{k=1}^G \Phi_k(1) = 2D (G - G^{(D-1)/D})$$

The approximated optimal average layout terminal-gate relationship becomes:

$$T \simeq t'_G \left(G - \frac{4(G - \sqrt{G})}{4} \right) + \frac{G}{G_c} T_{ext} = t'_G \sqrt{G} + \frac{G}{G_c} T_{ext}. \quad (4.62)$$

The square root in this expression guarantees that the slope of the complete expression in a logarithmic graph (i.e., the Rent's exponent) is never smaller than 0.5 (or $(D - 1)/D$, for a D -dimensional layout medium).

At the other end of the scale, the worst possible wire length distribution is that which corresponds to a random placement. In this case, $\mathcal{N}(\ell)$ equals the site distribution $\mathcal{D}(\ell)$ and the occupation probabilities $\mathcal{Q}_k(\ell) = \mathcal{Q}(\ell)$ are constant values, independent of ℓ or the position within the placement grid:

$$\mathcal{Q}(\ell) = \frac{N_c}{G_c(G_c - 1)/2} = \frac{t'_G}{G_c - 1}, \quad (4.63)$$

if we assume that the circuit size equals the size of the placement grid. This allows the direct calculation of Equation (4.46), for which we find that:

$$E_i \left[\frac{\mathcal{N}_{k,i}(\ell)}{\Phi_{k,i}(\ell)} \right] = \frac{1}{G_c - 1}, \quad (4.64)$$

and hence:

$$T = t'_G \left(G - \frac{\sum_{k=1}^G \sum_{\ell=1}^{\ell_{max}} \Phi_{g_k, \mathcal{G}}(\ell)}{G_c - 1} \right) + \frac{G}{G_c} T_{ext}. \quad (4.65)$$

In this expression, the summation across all wire lengths yields $G - 1$, i.e., the total number of gate locations in the region to which g_k can be connected. The second summation adds an extra factor G , so we find:

$$\begin{aligned} T &= t'_G \left(G - \frac{G(G - 1)}{G_c - 1} \right) + \frac{G}{G_c} T_{ext} \\ &= t'_G \frac{G(G_c - G)}{G_c - 1} + \frac{G}{G_c} T_{ext}. \end{aligned} \quad (4.66)$$

This expression is the same as the solution to Christie's differential equation for a random placement (4.30). As long as $G \ll G_c$, the number of terminals for geometric modules from a random placement scales approximately proportional to the module size.

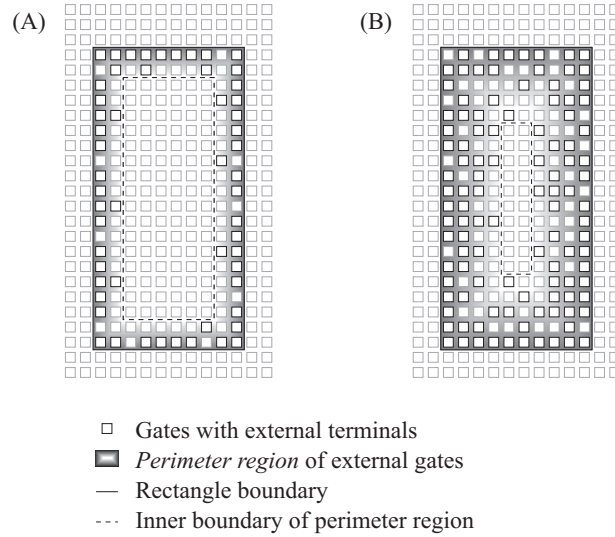


Figure 4.24: Schematic representation of the gate locations where the external terminals (for the module) originate, situated in a *perimeter region*. Figure (A): for low values of the partitioning Rent exponent this region is narrow, compared to the region side, and the region area is roughly proportional to the rectangle perimeter. Figure (B): for high values of $p_{(p)}$ this is no longer the case, because there are significantly more long wires.

For realistic wire length distributions, resulting from optimized placement, the region I scaling behavior lies somewhere between the perimeter scaling of Expression (4.62) and the area scaling of Expression (4.66). In fact, one might consider the number of terminals to be proportional to the number of gates in a *perimeter region* (Figure 4.24), from which the majority of the region's external terminals originate. The width of this region depends on the scaling behavior of the wire length distribution.

Additionally, the average number of terminals in the average layout Rent characteristic can never be less than that for the (topological) partitioning Rent characteristic. Hence, we can expect that $p_{(al)} \geq p_{(p)}$, apart from minor deviations caused by the fitting procedure.

Scaling behavior for small and large module sizes

The above approximations are only valid in the central part of the Rent characteristic (i.e., region I). There are boundary effects for the largest

module sizes (region II), as well as for the smallest module sizes (region III).

The presence of a region II is simply caused by the fixed number of external terminals in the circuit, which “pulls down” the high end of the average layout Rent characteristic. This effect, combined with the error margin of the fitting procedure, may cause fitted values of the average layout Rent parameter $p_{(al)}$ to turn out slightly below $1 - \frac{1}{D}$, in particular for small circuits or when there are few data points.

For very small region sizes (region III), the scaling behavior of average layout Rent characteristics is close to that for a random placement. Using the image of a perimeter region, this occurs as long as the module dimensions are small, compared to the width of the perimeter region: almost all gates in the region contribute to the module terminals. A more detailed explanation regarding the scaling behavior of average layout Rent characteristics for square and rectangular modules can be found in [DVSVC01]. Note that the concept of a region III of Rent’s rule was originally described for the partitioning Rent characteristic by Stroobandt in [Str99, Str98]. However, the effect we have just discussed is inherent to the layout interconnect behavior and thus probably complementary to the effect identified in [Str99, Str98]. Still, we will maintain the term “region III” of Rent’s rule to indicate any effects that are specific to small module sizes.

Impact of the region shape

In Davis’ technique for predicting wire length distributions, the same Rent parameters are applied to many different regions with a variety of different shapes, evolving from triangles to rectangles with different aspect ratios. In this section, we explore the impact of the shape of layout modules on the resulting Rent characteristic. In particular, for computational reasons, we will focus on rectangular regions with different aspect ratios χ , defined as their shortest side, divided by their longest side, both expressed in Manhattan units ($\chi \in]0, 1]$). We will consider Rent characteristics that are obtained from regions that all have the same aspect ratio.

In the previous chapter, the question was raised whether it is correct to apply the same Rent parameters to all of these regions. A simple intuitive argument raises the suspicion that it is not.

Let us consider a hypothetical netlist with gates that are all the same

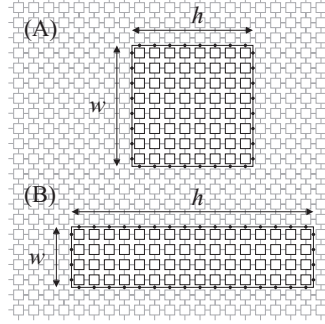


Figure 4.25: Schematic representation of a mesh circuit layout and the number of external terminals of rectangular regions with $G = 64$: (A) $w = 8$, $h = 8$, $\chi = 1$ and $T = 32$; (B) $w = 4$, $h = 16$, $\chi = 4$ and $T = 40$.

size and that are interconnected like the nodes of a regular square grid, i.e., a mesh (Figure 4.25). It is clear that the optimal layout for this circuit in the plane is a mesh structure. In this particular case, the relation between T and G can be calculated analytically for, e.g., rectangular subregions with width w and height $h = \chi w$. The number of terminals for these regions is given by

$$T = 2(h + w) = 2(1 + \chi)w, \quad (4.67)$$

while the number of gates equals

$$G = hw = \chi w^2. \quad (4.68)$$

This leads to

$$T = 2 \frac{1 + \chi}{\sqrt{\chi}} G^{0.5}. \quad (4.69)$$

Fitting the power law form of Rent's rule to Equation (4.69), results in:

$$\begin{cases} p = 0.5 \\ t = 2 \frac{1 + \chi}{\sqrt{\chi}} = t'_G \frac{1}{2} \frac{1 + \chi}{\sqrt{\chi}} \end{cases} \quad (4.70)$$

We find for this example that $p_{(al)}$ is independent of the aspect ratio χ , but $t_{(al)}$ is not. Even though the circuit in this model is not representative, we might expect to see a similar effect for real circuits.

The same result can be found from our model for the best possible wire length distribution (Equation (4.59)). For rectangular regions and

a two-dimensional placement grid, the reader can verify that Expression (4.61) becomes:

$$\sum_{k=1}^G \Phi_k(1) = 4 \left(G - \frac{1}{2} \frac{\chi + 1}{\sqrt{\chi}} \sqrt{G} \right), \quad (4.71)$$

yielding an average number of module terminals:

$$T = t'_G \frac{1}{2} \frac{\chi + 1}{\sqrt{\chi}} \sqrt{G} + \frac{G}{G_c} T_{ext}. \quad (4.72)$$

The expression for the number of terminals corresponding to a random placement does not change, since it only depends on the module size and not on the region shape.

Using our model (Equation (4.56)), we have calculated the average layout Rent characteristics for rectangular regions, as a function of the aspect ratio χ . For our experiments, we have used the wire length distributions from square grid placements, obtained with SA. Figure 4.20 shows some results for benchmarks q_{25} , q_{50} and q_{75} . It illustrates that indeed, the average layout Rent exponent is largely independent of the region shape. However, the entire Rent characteristic is shifted upwards as the aspect ratio moves further away from 1. For a more extensive discussion of the impact of the region aspect ratio on average layout Rent characteristics, we refer again to [DVSVC01].

4.3.4 Consequences for Davis' prediction technique

Now that we have determined some of the properties of average layout Rent characteristics, let us reconsider the core equation of Davis' prediction technique:

$$N_A(\ell) = \frac{1}{2} [(T_{AB} - T_B) - (T_{ABC} - T_{BC})] \quad (4.73)$$

$$= \frac{1}{2} [T(G_B + 1) - T(G_B) + T(G_{BC}) - T(G_{BC} + 1)] \quad (4.74)$$

$$= \frac{1}{2} t [(G_B + 1)^p - (G_B)^p + (G_{BC})^p - (G_{BC} + 1)^p]. \quad (4.75)$$

The second line in this equation expresses the assumption that a single terminal-gate relationship can be used for all module sizes and shapes, while the third line states that this terminal-gate relationship can be modeled by Rent's rule with some appropriate value of p .

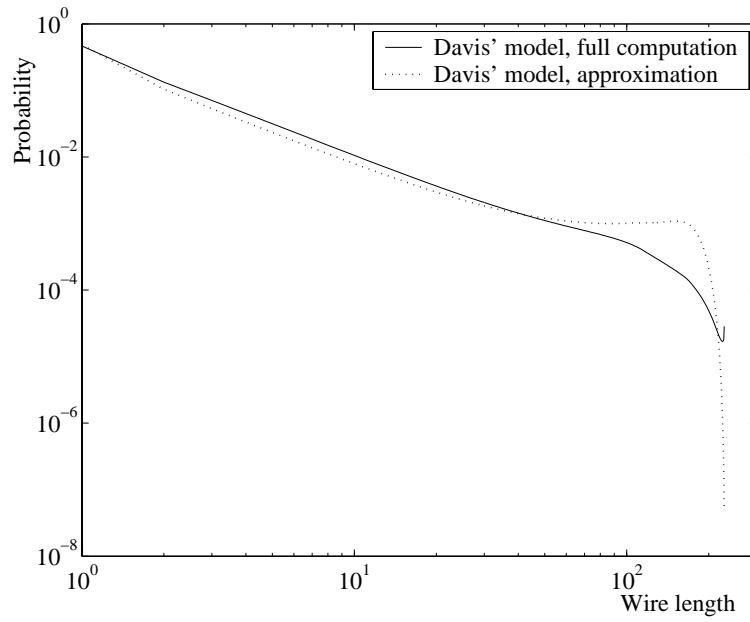


Figure 4.26: Predicted wire length distributions from Davis' full model and our Approximation (3.29) when using Christie's parametric expression (4.30) for the geometric Rent characteristic (the placement grid size, as well as t'_G , G_c and T_{ext} were taken from benchmark `ibm01` and $\rho = 0.7$).

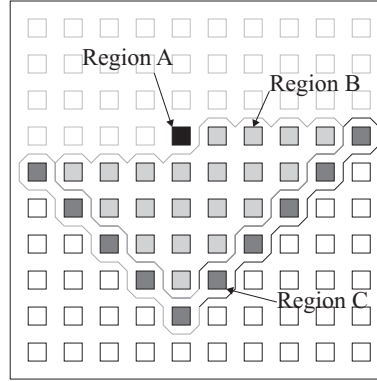


Figure 4.27: Regions *A*, *B* and *C* when traversing the layout grid in Davis' technique.

As we know, Rent's rule is definitely not an accurate model for any terminal-gate relationship. So let's make one step back and use, e.g. Christie's analytical terminal-gate relationship (4.30) in Equation (4.74). Figure 4.26 shows the result of using Christie's model in Davis' full model and Approximation (3.29). There appear to be two problems with these wire length distributions. The first is the large over-estimation of the number of unit-length wires. The slope (on logarithmic scale) of this distribution is even larger between $\ell = 2$ and $\ell = 1$ than for the rest of the wire lengths. This does certainly not accord with observations made from placement experiments. The second problem occurs for very long wires, where the distribution seems to "flip upwards" in Davis' full model. In the approximated version of Davis' model, both effects are even stronger.

It is our belief that both of these inaccuracies are due to the assumption that a single Rent characteristic may be applied for all module shapes and sizes. From our analysis of layout modules and their numbers of terminals, we have found that there are different effects that contradict this. There is an impact of the region shape as well as the region position on the number of module terminals. In particular, in Davis' full model, all regions are boundary regions, because the part of the grid that has already been visited is virtually eliminated (see Figure 4.27). Also, region *A* always contains a single *boundary* grid location. In a well-optimized placement, and for a sufficiently large region *B*, there can be expected to be more wires going from *A* to *B* than from *A* to the exterior of *B*. Hence, the region that consists of the union of *A* and *B*,

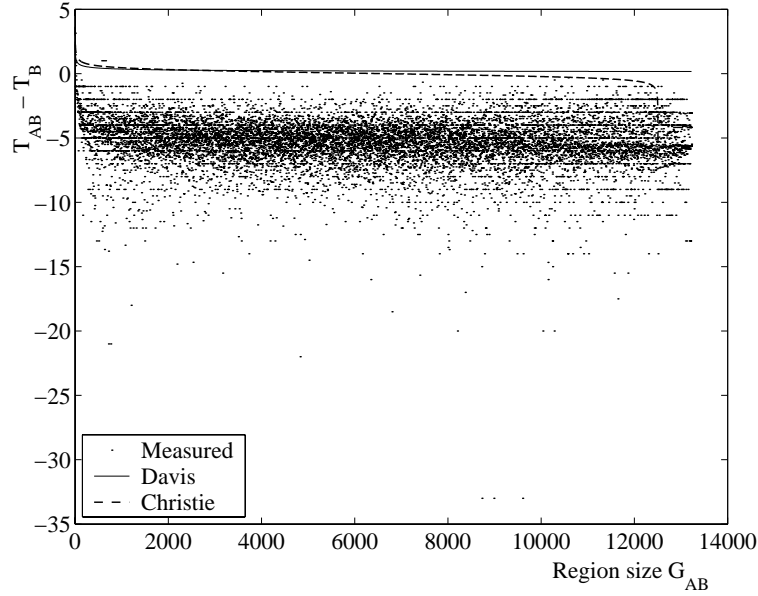


Figure 4.28: Measured values of $T_{AB} - T_B$ as a function of $G_B + 1$ when traversing the placement grid in Davis' interconnect prediction (data extracted from a SA placement of benchmark *ibm01* in a grid of 115×115 square cells). The full line indicates the value of $T_{AB} - T_B$ that results from using Rent's rule, while the dashed line shows the same for Christie's model for layout Rent characteristics (4.30).

will very often have fewer terminals than region B alone, even though it is larger. We have measured this for a SA placement of benchmark *ibm01*. Figure 4.28 shows the measured data points of the difference between T_{AB} and T_B that occurs in Equation (4.73). If a single terminal-gate relationship applies, this difference should be positive as long as the terminal-gate relationship is monotonically increasing (also shown in Figure 4.28 for Rent's rule and Christie's model). However, it was measured to be negative for most of the regions B and AB that occur in Davis' grid traversal. Also, this Figure shows that this difference has far from a constant value for each region size G . Instead, it varies enormously, with values ranging from just below zero to -10 and less.

Our results show that at least two different terminal-gate relationships would be required in Davis' model: one for regions B (or BC) and another for regions AB (or ABC). This implies that, even if a parameterized model for the average number of terminals of layout modules were to be found, and if the parameters in this model could be esti-

mated a priori, it still wouldn't result in a very high accuracy of Davis' model.

4.4 Comparative discussion

In this chapter, we have studied Rent characteristics and their usefulness for predicting interconnect properties.

The partitioning Rent characteristic, to be used in Donath's model, could already be measured a priori. Based on a model that was originally proposed by Verplaetse [Ver03], we have established a single-parameter model (henceforth called the β -model) that fits the measured Rent characteristics very well. Although this is definitely a topic for further research, this can be a first step towards the abstraction of Donath-based prediction techniques for trade-offs earlier in the design cycle (i.e., before the exact netlist is available).

With respect to geometric terminal-gate relationships, we have explored a parametric model that was proposed by Christie in [Chr01]. This model corresponds to measured data a lot better than Rent's rule, and our own extensions of to an even better match. However, it still offers no indication of how the model parameters are to be estimated a priori.

Based on our enumeration model for layout Rent characteristics, we can conclude that they are certainly different from the partitioning Rent characteristics. We have determined a theoretical lower bound to their scaling behavior: $p_{(al)} \geq 0.5$ for 2-D placement grids. This bound is different from that for partitioning Rent characteristics ($0 \leq p_{(p)} \leq 1$).

It has been shown that the number of terminals of a geometric module not only depends on region size, the circuit topology, the placement architecture and the placement quality, but also on the position of the region in the placement grid and on the region shape. Hence it is by no means correct to use a single Rent characteristic for all regions used in Davis' model for interconnect prediction. Moreover, since the terminal-gate relationships for layout modules are clearly a result of the wire length distribution, they are not available for a priori predictions. This makes Davis' approach altogether unsuitable for predicting wire length distributions for individual circuits.

Therefore, we can conclude from the results presented in this chapter that an interconnect prediction approach based on Donath's tech-

nique is by far the most promising to achieve the goals that were set in this Ph.D. thesis.

Chapter 5

Towards the accurate prediction of placement wire length distributions

In the previous chapter, we have determined that Donath's prediction technique is the best candidate to serve as a basis for a more accurate a priori prediction of interconnect properties. However, there are some intrinsic limitations to Donath's original technique. In this chapter, we systematically address most of these limitations to achieve a priori predictions of the wire length distribution and the average wire length. First, we address the limitations regarding the circuit graph model. We propose to use the partitioning Rent characteristic instead of Rent's rule. Second, we extend the architecture model to rectangular gridded architectures with rectangular cells and evaluate the resulting model across a wide range of architectural variants. To obtain predicted average wire lengths that are closer to the measured values, we evaluate two occupation probability functions, proposed by Stroobandt and Verplaetse, respectively. Finally the sensitivity of our model to variations in the Rent characteristic is investigated.

5.1 Using the Rent characteristic in Donath's model

In Chapter 3, the approximation of the Rent characteristic by Rent's rule was identified as the most probable cause of the bad correlation between predicted average wire lengths from Donath's model and experimentally measured values. If the real partitioning Rent characteris-

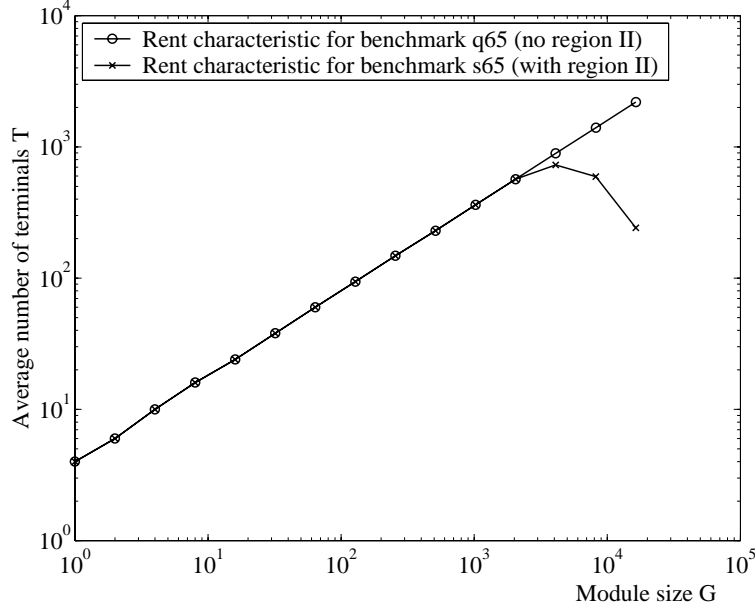


Figure 5.1: Partitioning Rent characteristic for benchmark circuits q65 and s65: both have the same size and partitioning Rent exponent $p = 0.65$, but q65 is strictly Rentian, while s65 has a region II.

tic $T(G)$ is available, it can easily be used in Donath's model. However, in this case, a numerical evaluation of the model is required. The number of nets cut at each hierarchy level, previously given by Equation 3.2:

$$N_k = t4^{(k-1)p}(2 - 2^{2p-1}), \quad (5.1)$$

now becomes:

$$N_k = \frac{4T(G_k/4) - T(G_k)}{2}. \quad (5.2)$$

In this Expression, $T(G)$ is the number of terminals given by (interpolation in) the partitioning Rent characteristic.

To evaluate the impact of using the Rent characteristic, we have generated a second series of synthetic benchmarks, s10-s90, with the same sizes and partitioning Rent exponents as the first series (q10-q90). However, instead of strictly adhering to Rent's rule, the partitioning Rent characteristics of these benchmarks show strong deviations for large module sizes, i.e., a pronounced region II (see Figure 5.1 for $p = 0.65$).

p	Predictions		Experiment	
	Rent's rule	Rent characteristic	(Plato-Quad)	
	$\mu(\ell)$	$\mu(\ell)$	$E[\ell]$	$stdev[\ell]$
0.10	2.37	2.33	2.33	0.010
0.15	2.53	2.47	2.46	0.016
0.20	2.72	2.78	2.78	0.018
0.25	2.96	3.04	3.04	0.022
0.30	3.25	3.35	3.35	0.030
0.35	3.61	3.66	3.66	0.039
0.40	4.07	4.14	4.14	0.058
0.45	4.63	4.97	4.97	0.080
0.50	5.34	5.83	5.83	0.119
0.55	6.22	6.97	6.97	0.172
0.60	7.32	8.79	8.79	0.262
0.65	8.67	10.91	10.90	0.388
0.70	10.33	13.91	13.89	0.580
0.75	12.33	18.25	18.22	0.882
0.80	14.72	24.07	24.03	1.334
0.85	17.51	32.36	32.30	2.021
0.90	20.70	44.26	44.17	3.078

Table 5.1: Average wire length results for synthetic benchmarks s_{10} - s_{90} (with region II): experimental values are average and standard deviation of the average wire length over 500 runs of Plato-quad, $\mu(\ell)$ is the estimated average wire length using Rent's rule and the Rent characteristic, respectively.

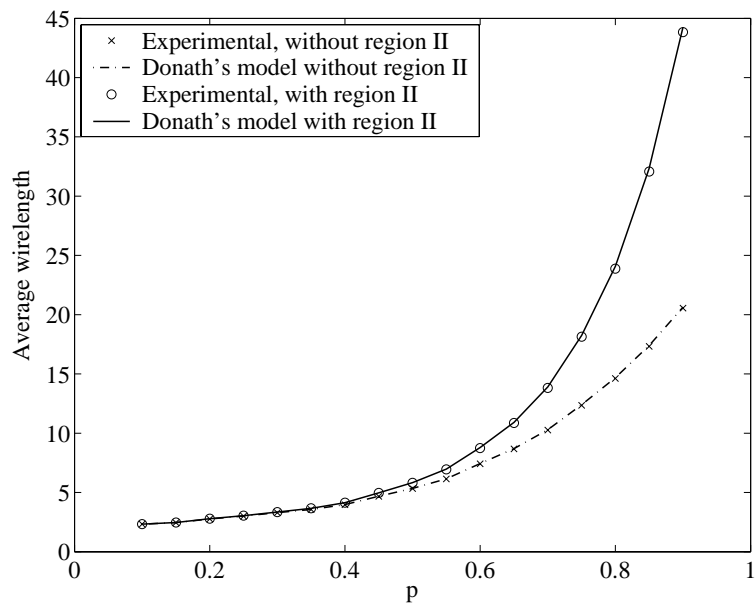


Figure 5.2: Average wire length for circuits without and with region II, as a function of the Rent exponent (experimental and model results for 4-way partitioning based placement).

The same experimental setup as in section 3.2.2 was used, leading to the results shown in table 5.1. Figure 5.2 shows the resulting average wire lengths as a function of the Rent exponent. They are clearly higher than the averages predicted by using Rent's rule in Donath's model, particularly for high values of the Rent exponent. This can easily be explained if we consider the number of nets connecting two submodules. For instance, for the case of 4-way partitioning, this number is equal to $N_k/6$. At the highest hierarchy levels, corresponding to region II, more nets are cut if a region II is present. Indeed, in Equation (5.2), N_k decreases when increasing the strength of region II, i.e., when reducing $T(G_k)$ while $T(G_k)$ is kept constant. This impact on the number of nets between two submodules is illustrated in figure 5.3. As a consequence, there are more long wires in the wire length distribution, which is reflected in a higher value of the average wire length. Finally, Figure 5.4 illustrates the impact of region II on the wire length distribution for a placement with `Plato-quad` of benchmark `s65`. Compared to an estimation based on Rent's rule, this figure clearly shows the increased number of long wires. This effect is predicted very accurately if the Rent characteristic is used in the model.

In contrast to what is assumed in Donath's model, in real circuit placements, the number of gates in the circuit usually doesn't match the number of gate positions in the layout grid, resulting in some empty grid positions after layout. Often, additional empty space is introduced to improve routability: if the gates are too close together, there is an increased risk of routing congestion. This is the case, e.g., in placement algorithms for FPGAs. Before we can validate using the Rent characteristic in Donath's model for the ISPD98 benchmarks, we need to determine how to compensate for this.

Consider an architecture with $X_g \times Y_g$ gate locations of which only $G_c < X_g Y_g$ are occupied. In our model, we assume the unoccupied gate locations (*the empty space*) in the placement grid to be uniformly distributed. Hence, an architecture module that consists of \tilde{G} grid locations will, on average, contain

$$\tilde{G} \frac{G_c}{X_g Y_g} \quad (5.3)$$

gates. As a result, in a partitioning-based placement model, its number of terminals must be given by:

$$\tilde{T}(\tilde{G}) = T\left(\frac{G_c}{X_g Y_g} \tilde{G}\right) \quad (5.4)$$

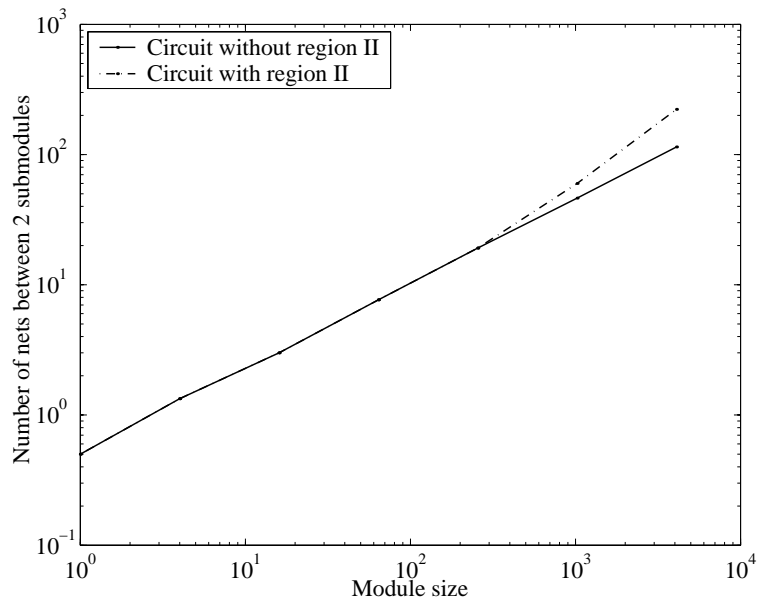


Figure 5.3: Number of nets between two submodules as a function of module size for two circuits with $p = 0.65$, one without region II (q65) and one with region II (s65)

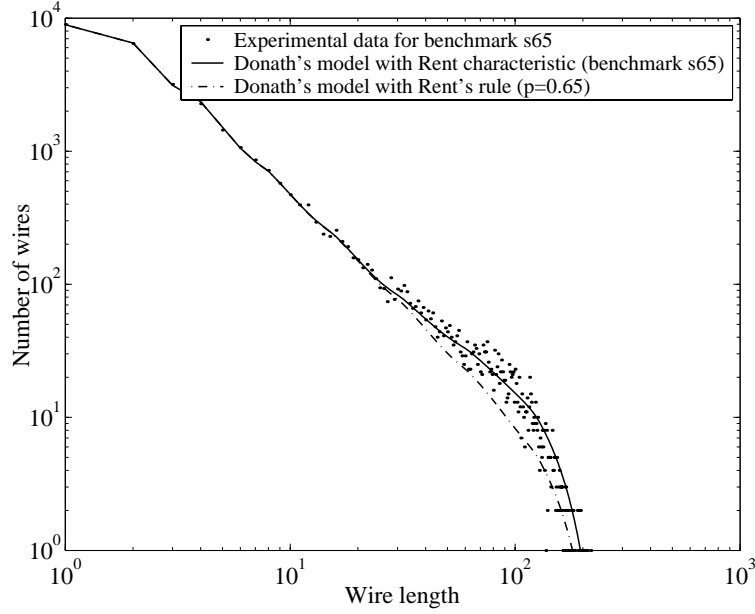


Figure 5.4: Theoretical and experimental wire length distribution for circuit s65 ($p = 0.65$ and $t = 4$).

The easiest way to incorporate this in our model is by *shifting* the entire Rent characteristic before starting our computations. This corresponds to the notion of inserting a number of dummy gates without terminals in the graph. Hence the average number of terminals for a single grid location is given by:¹

$$\tilde{T}(1) = t_G \frac{G_c}{X_g Y_g} \quad (5.5)$$

Note that this *shifting* transformation differs from the *stretching* transformation that was used in Section 4.2.

As we did in Section 3.2.2, we have used Donath's model to predict the average wire lengths for the ISPD98 benchmark circuits. The calculations were done for a square grid architecture of $2^H \times 2^H$ square cells with $H = \lceil \log_4(G_c) \rceil$. The resulting averages were then scaled with the factor η , given by Equation (3.10).

If we compare the results obtained by using the Rent characteristic

¹In reality, a gate location is either occupied, or it is not. However, experiments have shown the difference between this approximated computation and the full discrete computation to be negligible.

Benchmark	Experiment		Predictions with Donath's model	
	Plato-Quad	Rent's rule	Rent characteristic	Relative error (%)
ibm01	10.28	5.99	10.07	-2.04
ibm02	16.63	11.34	16.50	-0.81
ibm03	16.98	10.43	17.13	0.89
ibm04	15.71	10.61	15.45	-1.66
ibm05	22.15	12.91	21.88	-1.21
ibm06	18.87	11.11	18.66	-1.09
ibm07	17.20	9.90	17.00	-1.19
ibm08	20.43	12.73	20.18	-1.22
ibm09	15.35	8.00	15.06	-1.89
ibm10	17.34	9.62	16.97	-2.15
ibm11	16.76	9.60	16.44	-1.93
ibm12	21.04	11.21	20.72	-1.54
ibm13	17.46	10.11	17.18	-1.59
ibm14	21.36	14.25	20.86	-2.31
ibm15	21.74	10.84	21.36	-1.77
ibm16	22.04	12.59	21.62	-1.90
ibm17	24.69	11.04	24.94	1.03
ibm18	21.00	12.44	20.55	-2.13

Table 5.2: Average wire lengths resulting from Plato-Quad placements of the ISPD98 benchmark circuits and predicted values using Donath's model with Rent's rule and the partitioning Rent characteristic, respectively. The relative errors of this second prediction are also shown (global average relative error of -1.36%).

to the ones obtained from Donath's analytical model in Section 3.2.2, we can clearly see the large impact of using the Rent characteristic instead of Rent's rule (Table 5.2 and Figure 5.5). Because of the small wire length increase caused by the compaction algorithm in Plato-Quad, the predicted average wire lengths slightly underestimate the experimental results.

When compared to well-optimized placement results, the correlation between estimation and experimental results is clearly a lot better when the Rent characteristic is used. Obviously, the optimized placements result in far lower values of the average wire lengths, but the correlation between model and experiment remains very high. To illustrate this, Table 5.3 shows the correlation coefficients between the predicted average wire lengths obtained by using Rent's rule and the

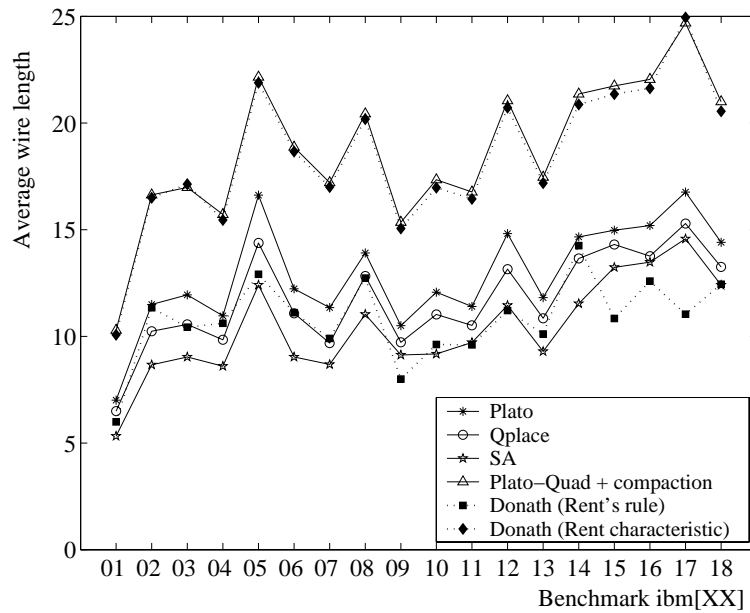


Figure 5.5: Average wire lengths resulting from different placement tools as well as predicted values obtained from Donath's model, using Rent's rule and the partitioning Rent characteristic, respectively.

	Plato-Quad	Plato	Qplace	SA
Plato-Quad		0.985	0.985	0.959
Plato			0.989	0.954
Qplace				0.954
Donath (Rent's rule)	0.800	0.804	0.788	0.697
Donath (Rent characteristic)	0.999	0.983	0.980	0.955

Table 5.3: Correlations between predicted average wire lengths and measured values obtained from different placement tools. The correlations between the different placement tool results are also shown.

Rent characteristic, respectively, and the experimental results.

Clearly, replacing Rent's rule in Donath's model by the partitioning Rent characteristic is essential for obtaining predictions that correlate well with experimental placement results. In fact, when compared to the results of `Plato-Quad`, a placement process that is very close to that which is actually modeled in Donath's model, this extension enables the accurate prediction of the average wire lengths. However, due to the intrinsic architectural restrictions, a scaling of the average wire length was required to obtain this average wire length. While this is relatively straightforward for square grid/square cell architectures, it is much less obvious when both the grid and the cells are allowed to be rectangular. The prediction of actual wire length distributions for such architectures seems entirely out of reach within the constraints of Donath's framework. In the next section, we propose a relaxation of those architectural constraints.

5.2 Relaxation of the architecture model

5.2.1 Model extensions

In Donath's original prediction model, the architectures were restricted to square grids of 4^H square cells. This was somewhat relaxed in [MY87] and [VM00, VMSVC95a, SVC97], but even in Van Marck's work, the number of cells in each direction must still be a power of 2. In both cases, the architecture is restricted because of the placement process that is mimicked. This restriction allowed the derivation of *analytical* models for the average wire length in both Donath's model as well as its extensions by Stroobandt [Str01b].

In this work, not only do we target the prediction of the average wire length, but also of the wire length distribution. To obtain this, even with Donath's model, a partial numerical evaluation is already required. Furthermore, we have shown in the previous section that the Rent characteristic must be used to obtain accurate results for individual circuits. Although the recursive partitioning model presented in Section 4.2 can be used for this terminal-gate relationship, it does not offer a closed-form analytical expression. Hence, with the extensions made thus far, a numerical model evaluation is inevitable. However, with the rapidly increasing computing power of ordinary desktop PC's, this poses no real problems. For example, our numerical (Matlab) implementation of Donath's model that uses the Rent characteristic takes less than a second (on a 500MHz PC), even for the largest benchmarks of the ISPD98 benchmark suite. Hence, there remain no arguments against a full relaxation of the underlying placement model.

Bipartitioning instead of four-way partitioning

To accord better with the reality of many partitioning-based placement tools, we first propose to use recursive bipartitioning instead of four-way partitioning. Van Marck already offered the choice between two-way, four-way or eight-way partitioning. For each module size, a partitioning strategy was selected depending on the relative average costs of wires in each direction, due to the different numbers of cells in each direction as well as different cell dimensions. In [DVSVC02], we showed that even for the four-way partitioning of a square grid, a model that uses recursive bipartitioning yields lower average wire lengths than a four-way partitioning placement model with the same Rent characteristic. Therefore, we always use bipartitioning in our models. The number of nets that is cut when partitioning a circuit module into two equal parts is now given by:

$$N_{cut,k} = \frac{2T(G_k/2) - T(G_k)}{2}. \quad (5.6)$$

Correction to allow for imperfectly balanced partitioning

The restriction that all dimensions of the placement grid must be powers of two originates from the requirement that each partitioning must be strictly balanced. This is only possible when every region dimension

that is cut consists of an even number of cells. As soon as the grid dimensions deviate from powers of two, odd dimensions for architecture modules must occur at some partitioning level. We will assume that the partitioning Rent characteristic, obtained by recursive bipartitioning, also offers a good approximation for a *slightly unbalanced* bipartitioning. For a module of size G that is partitioned into two modules of sizes G_a and G_b , respectively, we therefore replace Equation (5.6) by:

$$N_{cut} = \frac{T(G_a) + T(G_b) - T(G)}{2}. \quad (5.7)$$

To be able to compare our predicted results to the Plato-Quad placements, we also provide a 4-way partitioning version of our extended model. In this version, we calculate the number of nets that is cut as follows:

$$N_{cut} = \frac{\left(\sum_{i=1}^4 T(G_i) \right) - T\left(\sum_{i=1}^4 G_i \right)}{2} \quad (5.8)$$

Additionally, because the module partitioning now results in 4 submodules with potentially different dimensions, we assume that the number of nets connecting a submodule pair (a, b) with sizes (G_a, G_b) is proportional to the number of gates in both submodules:

$$N_{a,b} \sim \frac{G_a G_b}{\sum_{1 \leq i < j \leq 4} G_i G_j} \quad (5.9)$$

The architecture partitioning schedule

As a consequence of allowing unbalanced partitioning, the modules at a given hierarchy level no longer need to have equal sizes. Hence for small circuit modules, an overlap can occur between the smallest module sizes at a given hierarchy level and the largest module sizes at the level below. Therefore, we no longer use the notion of hierarchy levels.

Instead, we keep a (two-dimensional) list of module sizes, together with their frequency of occurring. Initially, this list only contains the entire placement grid. Then, at each step, the architecture module with the largest number of cells in either direction is selected for partitioning. For the bipartitioning version of our model, the decision whether to cut the X or the Y -direction is made as shown in Table 5.4 and the

Cut direction decision rules	
$(X > 1) \text{ and } (Y = 1)$	cut X
$(X = 1) \text{ and } (Y > 1)$	cut Y
$(X > 1) \text{ and } (Y > 1) \text{ and } (X_c X > Y_c Y)$	cut X
$(X > 1) \text{ and } (Y > 1) \text{ and } (X_c X < Y_c Y)$	cut Y
$(X > Y) \text{ and } (X_c X = Y_c Y)$	cut X
$(X < Y) \text{ and } (X_c X = Y_c Y)$	cut Y

Table 5.4: Decision rules for determining the cut direction when bipartitioning a module of $X \times Y$ grid locations with cell size $X_c \times Y_c$.

sizes of the submodules are computed such that the partitioning is as balanced as possible. The number of cut nets (for the total number of modules of that size) and their length distribution are calculated from Equation (5.7) and the wire length distribution is updated. Finally, the partitioned module is deleted from the module list and the new modules are inserted.

5.2.2 Validation of the extended model

With these extensions, we can estimate wire length distributions and average wire lengths for the `ispd98` benchmark circuits, placed in architectures with varying grid and cell dimensions. The full numerical version of our model was implemented in `Matlab`.² Some data regarding the run times of this implementation are reported at the end of this section.

Four-way and bipartitioning model versions

We have applied our full model, both with 4-way and bipartitioning, to predict the average wire lengths and wire length distributions for benchmarks `ibm01-ibm18`, for placements in square grids with square cells. Figure 5.6 and Table 5.5 compare the resulting average wire lengths to the average wire lengths obtained with different placement tools. The predicted values from the 4-way partitioning model version are very close to the values shown in Table 5.2 (column 4). Further-

²This implementation also supports 3-dimensional placement architectures. However, because we have not yet performed an experimental validation of the predictions for those architectures, no such predicted wire length distributions are reported here.

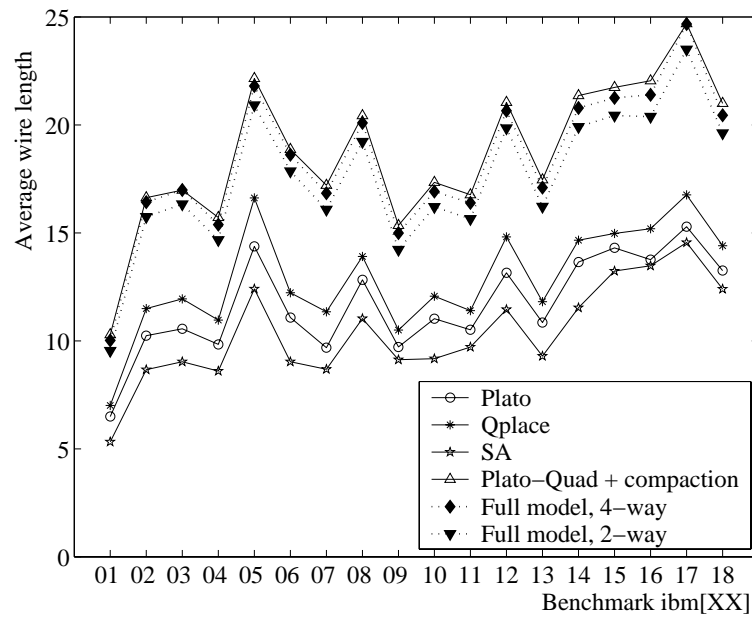


Figure 5.6: Average wire lengths for square grid/square cell placements, resulting from different placement tools as well as predicted values obtained from the bipartitioning and 4-way partitioning versions of our full model.

Benchmark	Experiments			Predictions with full model	
	Plato	Qplace	SA	4-way part.	2-way part.
ibm01	7.01	6.49	5.32	10.07	9.54
ibm02	11.50	10.24	8.66	16.50	15.74
ibm03	11.94	10.56	9.03	17.13	16.33
ibm04	10.96	9.84	8.60	15.45	14.67
ibm05	16.62	14.37	12.40	21.88	20.92
ibm06	12.23	11.08	9.03	18.66	17.86
ibm07	11.35	9.69	8.69	16.99	16.08
ibm08	13.91	12.82	11.03	20.18	19.22
ibm09	10.50	9.72	9.13	15.06	14.22
ibm10	12.07	11.03	9.17	16.97	16.20
ibm11	11.40	10.51	9.71	16.45	15.65
ibm12	14.81	13.15	11.45	20.72	19.85
ibm13	11.81	10.85	9.29	17.19	16.22
ibm14	14.66	13.65	11.54	20.86	19.91
ibm15	14.97	14.30	13.23	21.36	20.44
ibm16	15.19	13.76	13.48	21.62	20.38
ibm17	16.76	15.29	14.57	24.94	23.49
ibm18	14.41	13.25	12.40	20.55	19.62

Table 5.5: Average wire lengths for ISPD98 benchmarks: experimental values obtained with `Plato` (average of 10 placement runs), `Qplace` and `SA`, as well as predicted values obtained by our adapted version of Donath’s model. The results shown are for the 4-way and bipartitioning model versions (without occupation probability).

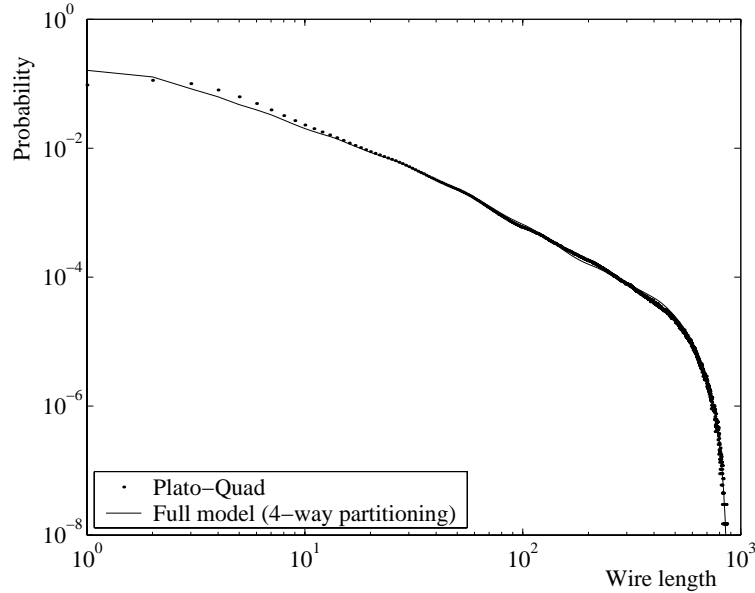


Figure 5.7: Measured wire length distribution for benchmark `ibm17` (Plato-Quad) and predicted distribution using our 4-way partitioning model version.

more, the predicted wire length distribution matches the measured distribution very well, particularly for the long wires (see Figure 5.7). Hence, we can conclude that our full model with four-way partitioning accurately predicts the Plato-Quad placement results.

Figure 5.6 also confirms that the predicted averages resulting from the bipartitioning model are lower than those from the four-way partitioning model. Henceforth, we will only refer to bipartitioning model results.

Table 5.6 (first two columns) shows the correlations between the predicted average wire lengths from both of these model versions and the averages obtained with Plato, Qplace and SA, respectively. They are comparable to the correlations between the results of the different tools that were given in Table 5.3. Table 5.6 also shows the average, minimal and maximal relative errors on the predicted average wire lengths when compared to the results of each placement tool. Obviously, our predictions are an over-estimation, since they model a placement process with only very limited placement optimization. However, considering the high correlation values, the estimations can certainly be used for exploration purposes.

		Without Occ. prob.		With Occ. prob.	
		4-way	2-way	Stroobandt	Verplaetse
Error values:					
Plato	Avg. (%)	43.50	36.47	-50.41	6.57
	Min. (%)	31.70	25.87	-58.99	-0.07
	Max. (%)	52.56	46.03	-41.09	14.50
Qplace	Avg. (%)	58.20	50.45	-45.26	17.50
	Min. (%)	49.33	42.89	-57.08	9.74
	Max. (%)	75.47	66.02	-33.87	29.74
SA	Avg. (%)	79.72	70.94	-37.65	33.52
	Min. (%)	60.42	51.24	-53.61	18.39
	Max. (%)	106.60	97.75	-21.84	55.06
Correlation values:					
Plato		0.983	0.985	0.887	0.987
Qplace		0.980	0.982	0.840	0.979
SA		0.955	0.952	0.740	0.943

Table 5.6: Average, minimal and maximal relative deviations $100 \times (\mu_{\ell, \text{pred}} - \mu_{\ell, \text{meas}}) / \mu_{\ell, \text{meas}}$ of predicted average wire lengths, compared to measured average wire lengths (data from Tables 5.5 and 5.9) across all ISPD98 benchmarks. The correlations between predicted and measured average wire lengths across all ISPD98 benchmarks are also shown.

Benchmark	Full model (no occ. prob.)	Full model (occ. prob. Verplaetse)
ibm01	0.964	0.957
ibm02	0.990	0.987
ibm03	0.999	0.997
ibm04	0.987	0.982
ibm05	0.998	0.996
ibm06	0.979	0.983
ibm07	0.993	0.995
ibm08	0.993	0.994
ibm09	0.998	0.998
ibm10	0.999	0.997
ibm11	0.999	0.999
ibm12	0.993	0.992
ibm13	0.995	0.995
ibm14	0.996	0.994
ibm15	0.992	0.988
ibm16	0.998	0.996
ibm17	0.989	0.991
ibm18	0.988	0.990

Table 5.7: Correlation values between predicted average wire lengths and `Plato` placement results across 70 architectural variants.

Exploration of architectural variants

To validate the usefulness of our model (bipartitioning version) for the exploration of layout and technology options, we have applied it to a range of architectures for each of the ISPD98 benchmarks. The architectures were the same as those used for the evaluation of Davis' prediction technique in Chapter 3, i.e., 10 grid aspect ratio's ranging from 0.1 to 1 (square) and seven cell aspect ratios ($\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, 1, $\frac{4}{3}$, $\frac{3}{2}$ and 2) with equal cell areas. For each of the 70 architectural variants, 10 placement runs with `Plato` were performed and the wire length distribution was averaged across the 10 results.

The observed correlations between predicted and measured average wire lengths (Table 5.7) are again very high. Figure 5.8 shows how the average wire length changes as a function of the grid and cell aspect ratios. The predicted values (also shown in the Figure) follow that trend.

Finally, Figure 5.9 shows an example of a measured and predicted

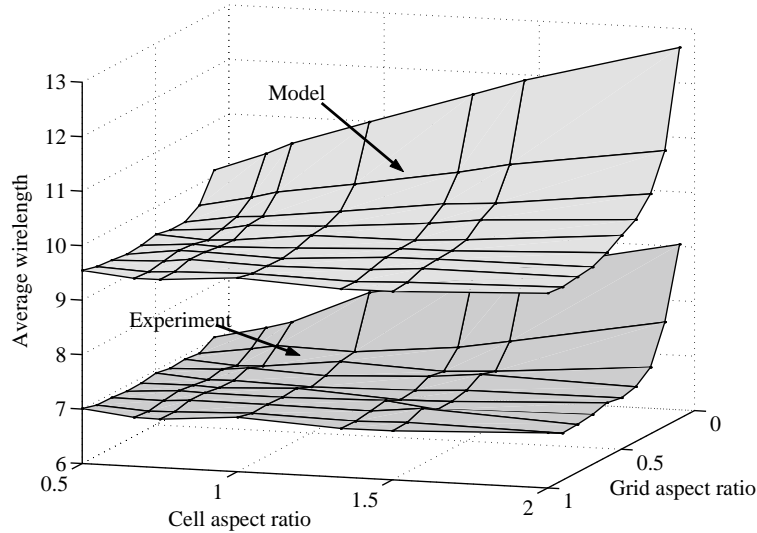


Figure 5.8: Exploration of the impact of grid aspect ratio and cell aspect ratio on the average wire length for ISPD98 benchmark *ibm01* (results from our full bipartitioning model without occ. prob. and experimental results).

wire length distribution for benchmark *ibm17* (grid aspect ratio of $\simeq 0.2$). The dotted line represents the predicted wire length distribution from our full model (bipartitioning version).

Some implementation and run time aspects

Our implementation was not heavily optimized for speed, but it is still very fast. By way of example, Table 5.8 reports some run times of our full prediction model on a 500MHz Pentium II processor with 194MB RAM. Columns 2-4 show some details for a single prediction (square grid/square cell placement). Because our implementation performs the calculations only once for each module size, the run times depend strongly on the number of different module sizes that occur in the partitioning tree. This number is minimal for a Donath-type square grid (sides equal to a power of two). As is shown in column 3 of Table 5.8, it is not a regular function of the grid side.

We have also measured the total run times for all 70 architectural variants described above (last column of Table 5.8). Because these have different grid sizes, the irregularities are somewhat averaged out for each benchmark. A power law fit of the run time data as a function

Benchmark	Square grid/square cell placement			Architectural variants
	X_g	No. of modules	Run time (s)	Run time (s)
ibm01	115	50	0.92	43.8
ibm02	143	54	0.89	42.8
ibm03	156	40	0.65	47.5
ibm04	170	48	0.80	50.9
ibm05	173	54	0.94	53.3
ibm06	185	54	0.96	57.5
ibm07	220	44	0.78	58.9
ibm08	232	38	0.68	59.5
ibm09	237	58	1.19	63.8
ibm10	269	66	1.34	72.1
ibm11	272	38	0.74	65.1
ibm12	273	68	1.40	71.8
ibm13	297	66	1.42	75.1
ibm14	394	64	1.41	96.3
ibm15	412	54	1.21	109.6
ibm16	439	66	1.81	121.9
ibm17	441	68	1.92	110.7
ibm18	471	66	1.93	119.3

Table 5.8: Run times of our full prediction model on a 500MHz Pentium II processor with 194MB RAM: prediction of the wire length distributions corresponding to a single square grid/square cell placement as well as for all 70 architectural variants reported in this section.

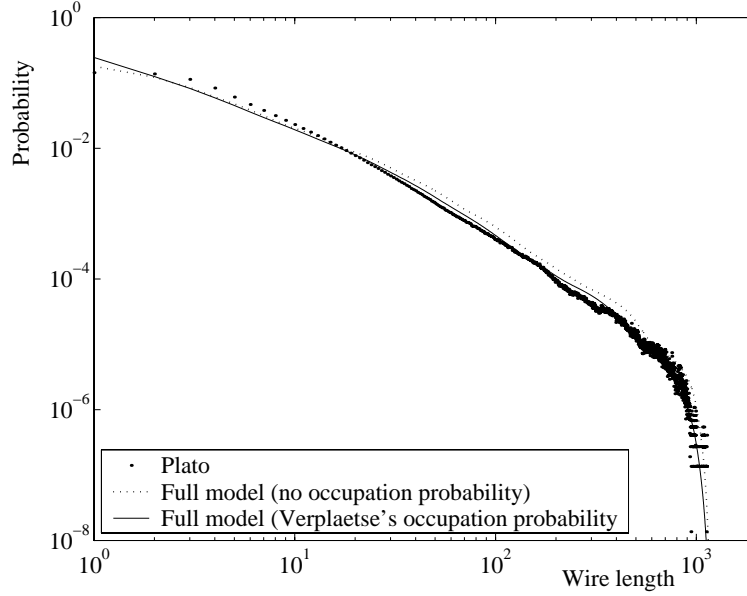


Figure 5.9: Measured wire length distribution for benchmark `ibm17`, placed with `Plato` in a grid of 987×189 square cells (grid aspect ratio $\simeq 0.2$) and predicted distribution using our bipartitioning model version.

of the circuit size yielded $\text{run time} \sim 0.92G_c^{0.39}$. A similar power-law fit to the square grid run times yields a slightly lower exponent (approximately 0.32). Some of the most time consuming operations in our implementation are the calculation of the site functions (convolutions) and the interpolation in the Rent characteristic.

5.3 Including better placement optimization

5.3.1 Stroobandt's occupation probability

So far, in all of the models described in this chapter the only placement optimization consists of the optimal recursive circuit partitioning. However, even in real partitioning-based placement tools, much more optimization is performed. As discussed in Chapter 3, Stroobandt [Str01b] proposed to incorporate a better placement optimization in Donath's model by introducing the occupation probability function:

$$Q(\ell) \sim \ell^{2p-4}. \quad (5.10)$$

Benchmark	Experiments			Full model + Occ. prob.	
	Plato	Qplace	SA	Stroobandt	Verplaetse
ibm01	7.01	6.49	5.32	3.17	7.31
ibm02	11.50	10.24	8.66	6.77	12.49
ibm03	11.94	10.56	9.03	6.07	12.76
ibm04	10.96	9.84	8.60	5.64	11.50
ibm05	16.62	14.37	12.40	8.85	16.61
ibm06	12.23	11.08	9.03	6.85	14.00
ibm07	11.35	9.69	8.69	5.79	12.57
ibm08	13.91	12.82	11.03	7.53	15.13
ibm09	10.50	9.72	9.13	4.37	10.93
ibm10	12.07	11.03	9.17	5.65	12.59
ibm11	11.40	10.51	9.71	5.53	12.18
ibm12	14.81	13.15	11.45	7.51	15.56
ibm13	11.81	10.85	9.29	5.93	12.69
ibm14	14.66	13.65	11.54	7.53	15.63
ibm15	14.97	14.30	13.23	6.14	15.70
ibm16	15.19	13.76	13.48	7.16	15.96
ibm17	16.76	15.29	14.57	7.36	18.23
ibm18	14.41	13.25	12.40	7.28	15.37

Table 5.9: Average wire lengths for ISPD98 benchmarks: experimental values obtained with `Plato` (average of 10 placement runs), `Qplace` and `SA`, as well as predicted values obtained by our adapted version of Donath’s model. The results shown are for the model with occupation probability according to Stroobandt and Verplaetse (bipartitioning model).

At each hierarchy level, the local site function is multiplied with $\mathcal{Q}(\ell)$ to obtain the length distribution of the wires connecting each submodule pair:

$$\mathcal{N}(\ell) \sim \mathcal{Q}(\ell) \Phi(\ell) \quad (5.11)$$

This occupation probability has always been considered to model *the optimal placement result*.

The concept of an occupation probability function was incorporated in our model. The default setting, $\mathcal{Q}(\ell) \sim 1$, corresponds to the assumption made in Donath’s model that the gates each wire is connected to are uniformly distributed across the respective submodules. However, we also allow the use of Stroobandt’s occupation probability (5.10). Note that the (fitted) Rent exponent p is used in this Equation.

Table 5.9 shows the average wire lengths predicted with this occupation probability for square grid/square cell placements of the ISPD98

benchmarks. The corresponding correlations with the three placement tools as well as the minimal, maximal and average relative errors are included in Table 5.6.

It is clear from these results that Stroobandt's occupation probability, in addition to the partitioning optimization that was already incorporated in the model, leads to very large under-estimations of the average wire length. In the results presented in [Str01b], the occupation probability (5.10) was applied in Donath's original model, which we now know to yield lower average wire lengths than our full model. However, on average, the under-estimation is smaller in these results. There are several possible causes for this difference. A first possible cause of difference is the fitting algorithm used for obtaining the Rent exponent p . It is possible that our fitting algorithm systematically yields lower p -values than that used by Stroobandt. This again illustrates the fact that Rent's rule is not sufficiently accurate. Due to its power-law nature, any prediction model that is based on Rent's rule is very sensitive to the fitted value of p .

A second possible cause for the smaller under-estimation in [Str01b] is the fact that they were obtained by very slow simulated annealing and that the benchmarks used were relatively small (up to 1200 gates). Possibly, due to the small problem size, these placements were far better optimized. Still, the fact that in our results the under-estimation is quite large, even compared to the SA placement results leads us to believe that the under-estimation resulting when using Stroobandt's occupation probability is probably more fundamental. We have found that it is very important to use the Rent characteristic instead of Rent's rule for calculating the number of nets that is cut when partitioning a module. Possibly, the Rent characteristic should also be used for calculating the occupation probability. However, how this is to be done is not at all clear and definitely a topic for further research.

5.3.2 Verplaetse's occupation probability

Whereas Stroobandt's occupation probability was intended to model the result of an optimal placement, Verplaetse [VDSVC01, Ver03] went in search of better models for the results of specific placement tools. He made a distinction between *flat* and *partitioning-based* placement approaches. In particular, he proposed an alternative occupation probability to model the impact of *swapping optimization*. In this optimiza-

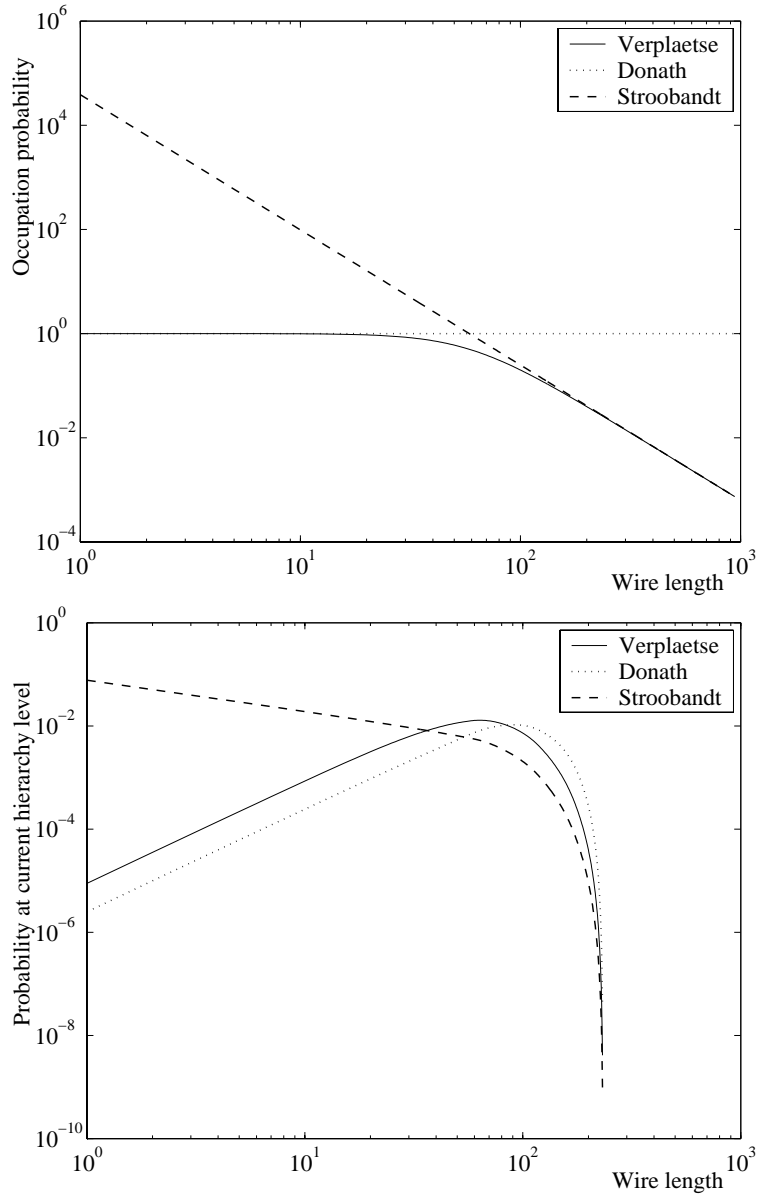


Figure 5.10: (top) Occupation probabilities used by Donath ($Q(\ell) \sim 1$), Stroobandt (5.10) and Verplaetse (5.12) for a Rent exponent $p = 0.65$ for a module of size 128×128 ; (bottom) Corresponding local wire length distribution when performing a 4-way module partitioning.

tion step, instead of randomly assigning the modules at each placement hierarchy level, the different module assignments are explored to optimize global placement result. As a result, there is some degree of wire length optimization within each hierarchy level. However, this optimization will mainly affect the longest wires at that level, because these dominate the cost function. Verplaetse suggested that this could be modeled by a local occupation probability that depends on the size of the module that is partitioned:

$$\mathcal{Q}(\ell, X) \sim \frac{1}{1 + \left(\frac{\ell}{X/2}\right)^{4-2p}}, \quad (5.12)$$

where $X \times X$ is the size of the module being partitioned. Figure 5.10 compares the occupation probabilities used by Donath ($\mathcal{Q}(\ell) \sim 1$), Stroobandt (5.10) and Verplaetse (5.12), as well as the resulting level-specific wire length distributions for $p = 0.65$ and $X = 128$.

To incorporate Verplaetse's occupation probability in our model, we need to adapt it to our relaxed architecture and partitioning constraints. Supposing that X is the dimension being cut, we propose to use the occupation probability function:

$$\mathcal{Q}(\ell, X, X_c) \sim \frac{1}{1 + \left(\frac{\ell}{(X_c X)/2}\right)^{4-2p}}. \quad (5.13)$$

A similar expression is used when partitioning the Y -dimension.

In `Plato`, most of the additional optimization (apart from the partitioning optimization) is obtained by swapping (see Appendix A). Therefore, our model with Verplaetse's occupation probability should offer relatively accurate predictions for the results of this placement tool. The average wire lengths obtained by our model when using Expression (5.13) for the occupation probability are included in Table 5.9 and plotted in Figure 5.11, while the resulting average, minimal and maximal relative errors, as well as the correlation between predicted and experimentally obtained values across different benchmark circuits are included in Table 5.6. In fact, when compared to `Plato`, all predicted averages but one (benchmark `ibm06`) are within 10%.

The observed correlations between predicted and measured average wire lengths are almost the same as those obtained without occupation probability. Still, as was the case with Stroobandt's occupation probability, it would probably be better to try to find an expression that relates to the Rent characteristic instead of Rent's rule.

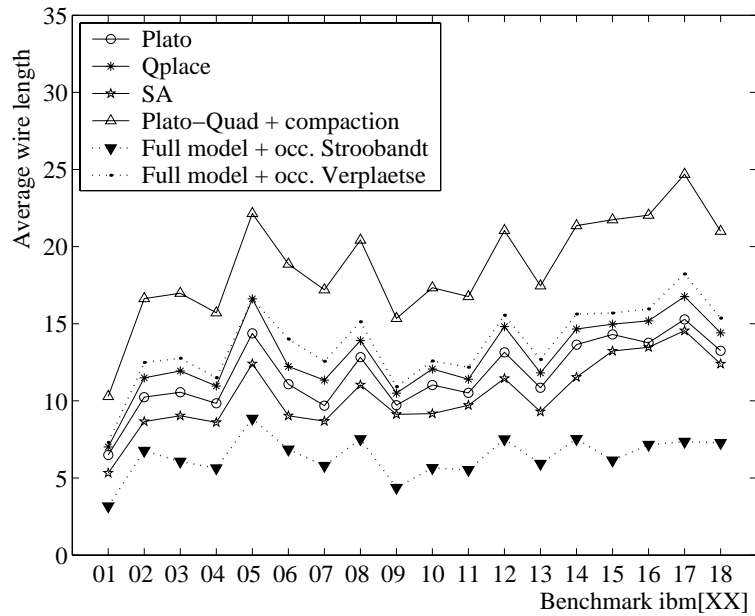


Figure 5.11: Average wire lengths for square grid/square cell placements, resulting from different placement tools as well as predicted values obtained from the bipartitioning version of our model, combined with the occupation probabilities of Verplaetse (5.13) and Stroobandt (5.10).

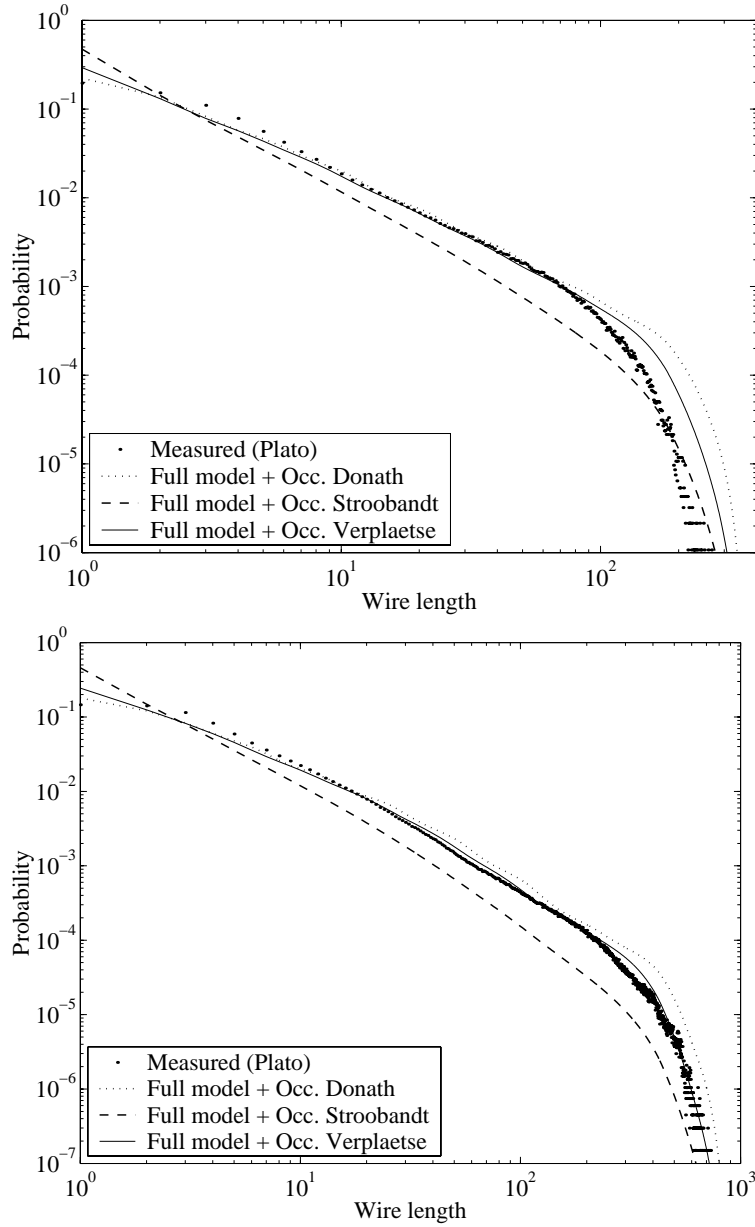


Figure 5.12: Wire length distributions obtained by `Plato` and with our full bipartitioning model, combined with the three occupation probabilities: Donath's (no occ. prob.), Stroobandt's (5.10) and Verplaetse's (5.13): (top) Benchmark `ibm06`, which shows the largest observed deviation between predicted and measured wire length distribution, and (bottom) benchmark `ibm17`, which is a typical representative for the rest of the benchmarks.

Figure 5.12 compares the wire length distribution obtained with `Plato` to the predicted distributions from our model, using the three occupation probabilities discussed so far (benchmarks `ibm06` and `ibm17`). Benchmark `ibm17` is a typical representative for most of the benchmarks in the ISPD98 series. Its number of long wires is over-estimated when no occupation probability is used and largely under-estimated when using Stroobandt's occupation probability. However, when using Verplaetse's occupation probability, there is a close to perfect match. The largest remaining errors lie in the length range from $\ell = 1$ to (approximately) $\ell = 10$. The predicted wire length distributions for non-square grids and cells are equally close to the measured ones. As an example, the distribution for a grid with aspect ratio 0.2, predicted by using Verplaetse's occupation probability, is also shown in Figure 5.9. The last column in Table 5.7 shows the correlation of the predicted average wire lengths across 70 architectural variants to the measured ones. They are comparable to those obtained without occupation probability.

So far, we have not been able to pinpoint the reason for the quite large observed deviation for benchmark `ibm06`. The fact that Rent's rule is used for the occupation probability is a possible cause. However, the quality of the Rent's rule fit for this benchmark was not observably different or worse than that for many other benchmarks in the ISPD98 benchmark series. In [Ver03], the impact of heterogeneity in circuit complexity was studied. It is possible that this circuit contains parts with very high and other parts with very low complexity. When no occupation probability is used, the variability of the number of terminals per gate is averaged out in a linear way in our model. However, the power-law form of the occupation probability is strongly non-linear. Both Verplaetse's and Stroobandt's occupation probability use the Rent exponent of the *average* terminal-gate relationship. This approximation turns out to be rather good for most benchmarks. However, for very heterogeneous circuits, it would probably be better to try to *separate* the parts with strongly different complexities, determine the Rent exponent for each of them and perform separate calculations for each part. Still, the determination of the true cause of the bad prediction for benchmark `ibm06` remains a topic for further research.

Figure 5.13 shows the three predicted wire length distributions from Figure 5.12 (benchmark `ibm17`), but now compared to the wire length distribution obtained with `SA`. In Verplaetse's work, it is also suggested that a *flat* prediction model such as Davis' might be better to predict

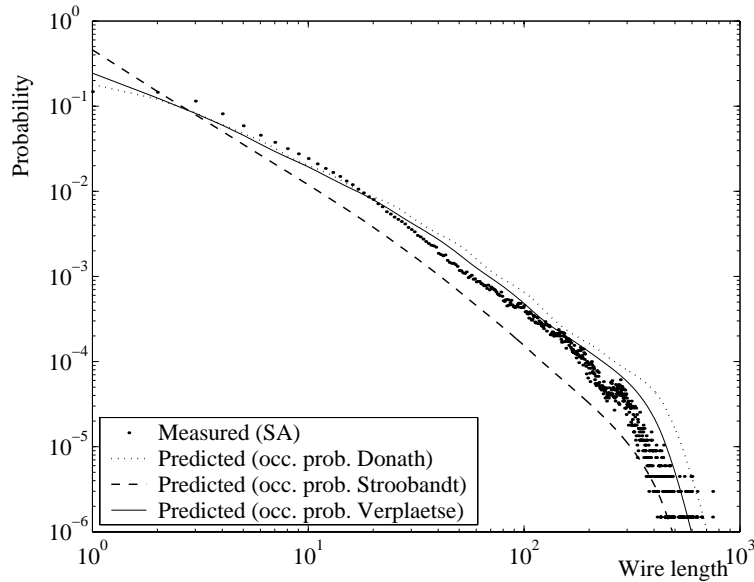


Figure 5.13: Comparison of the wire length distribution resulting from a square grid/square cell SA placement of benchmark `ibm17` to the predicted distributions, obtained by using the three occupation probability functions discussed in this section.

the results of flat placement tools (such as SA). Remember that Davis' model could be approximated by multiplying the site function of the entire placement grid with an occupation probability that has the same expression as Stroobandt's. The difference is that Davis applies his occupation probability in a *flat* way (i.e., to the entire placement grid at once), whereas Stroobandt applies it in a hierarchical fashion. Figure 5.12 clearly shows that the wire length distribution obtained by Stroobandt's model is still a gross over-estimation of the probabilities for short wire lengths. We can therefore expect that a global application of the same occupation probability will also lead to under-estimations.

The difference between the wire lengths from the `Plato` and SA placements seems to be strongest for very long wires. Hence, a possible alternative to Stroobandt's occupation probability might be to use an expression similar to Equation 5.13, but without the adaptation to the module size. A more accurate prediction of the results for individual placement tools might be obtained by combining a flat and a hierarchical occupation probability. Obviously, a lot of work remains to be done to further refine the prediction models proposed in this section.

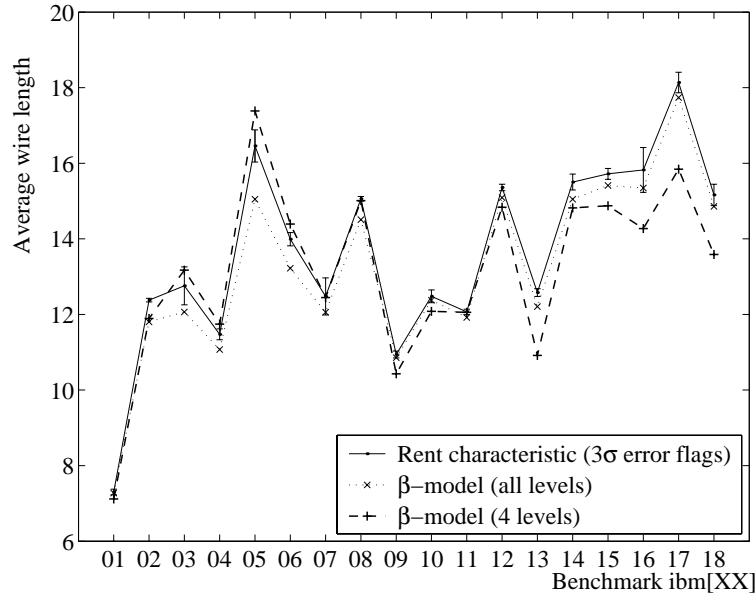


Figure 5.14: Sensitivity of the predicted average wire length to the Rent characteristic in our full model with Verplaetse’s occupation probability: average 10 instances of the measured partitioning Rent characteristic. The 3σ error flags are also shown for the predictions using the Rent characteristic.

5.4 Sensitivity of prediction quality to the Rent characteristic

To conclude this chapter, we discuss the sensitivity of our models to the Rent characteristic. First, we have generated 10 partitioning Rent characteristics for each of the ISPD98 benchmark circuits (each with a different random seed). Each of these was used in our full (bipartitioning) model with Verplaetse’s occupation probability. Then, we have fitted the cut probability model (4.23) from Section 4.2.2 (henceforth called the β -model) to each Rent characteristic, using 3, 4 and all partitioning levels, respectively. Table 5.10 shows the resulting mean and standard deviation of the predicted average wire length for all benchmarks.

The solid line in Figure 5.14 shows the average model results when the Rent characteristic is used, as well as the 3σ error flags. We can conclude that, for most benchmarks, the sensitivity to the randomized nature of the partitioning tool used for measuring the Rent characteristic is not very large.

5.4 Sensitivity of prediction quality to the Rent characteristic 175

Benchmark	Rent characteristic	β -model (all levels)	β -model (4 levels)	β -model (3 levels)
$E[\mu_\ell]$ - values:				
ibm01	7.30	7.27	7.12	7.09
ibm02	12.38	11.80	11.89	11.38
ibm03	12.76	12.07	13.17	13.47
ibm04	11.47	11.07	11.74	11.78
ibm05	16.46	15.05	17.39	17.77
ibm06	13.99	13.22	14.39	14.59
ibm07	12.48	12.06	12.45	12.53
ibm08	15.06	14.51	15.01	14.99
ibm09	10.94	10.85	10.43	10.47
ibm10	12.48	12.38	12.08	11.93
ibm11	12.07	11.92	12.06	12.06
ibm12	15.36	15.09	14.84	15.14
ibm13	12.58	12.21	10.91	10.65
ibm14	15.50	15.05	14.82	14.49
ibm15	15.72	15.41	14.88	14.85
ibm16	15.82	15.35	14.27	13.62
ibm17	18.14	17.74	15.84	16.03
ibm18	15.16	14.86	13.59	13.21
$Stdev[\mu_\ell]$ - values:				
ibm01	0.025	0.021	0.029	0.027
ibm02	0.016	0.021	0.024	0.021
ibm03	0.168	0.090	0.207	0.207
ibm04	0.048	0.027	0.062	0.074
ibm05	0.142	0.079	0.183	0.229
ibm06	0.059	0.040	0.059	0.093
ibm07	0.163	0.188	0.288	0.376
ibm08	0.021	0.025	0.056	0.052
ibm09	0.030	0.021	0.031	0.045
ibm10	0.056	0.053	0.090	0.115
ibm11	0.023	0.020	0.032	0.050
ibm12	0.029	0.029	0.054	0.062
ibm13	0.035	0.055	0.090	0.099
ibm14	0.070	0.069	0.110	0.119
ibm15	0.048	0.088	0.149	0.185
ibm16	0.198	0.225	0.467	0.440
ibm17	0.090	0.076	0.233	0.161
ibm18	0.094	0.091	0.137	0.143

Table 5.10: Sensitivity of the predicted average wire length to the Rent characteristic in our full model with Verplaetse's occupation probability: average and standard deviation across 10 instances of the measured partitioning Rent characteristic.

	Plato	Qplace	SA
Rent characteristic	0.986	0.979	0.943
β -model (all levels)	0.976	0.981	0.958
β -model (4 levels)	0.938	0.908	0.830
β -model (3 levels)	0.912	0.876	0.794
Rent's rule	0.804	0.788	0.697

Table 5.11: Correlations between the measured average wire lengths from Plato, Qplace and SA and the predicted average wire lengths obtained by our full model (with Verplaetse's occupation probability) when using the Rent characteristic, a β -model or Rent's rule.

However, the deviations from the Rent characteristic that are present in the β -model are sufficiently large to cause significant deviations in our prediction results. For several of the ISPD98 benchmarks, this deviation is very sensitive to the number of partitioning levels that is used for fitting. The stability of the predictions from the β -model is similar to that obtained with the Rent characteristic when all partitioning levels are used for fitting the cut probabilities. However, for some benchmarks, the standard deviation for the β -model results increases drastically when only a limited number of partitioning levels is used.

The main objective in introducing parametric models for the Rent characteristic is to allow the abstraction of interconnect prediction models to higher levels in the design hierarchy. In the past (with Donath's model) this was done by replacing a measured value of the Rent exponent by a typical one. In this context, our β -model certainly promises higher accuracy than Rent's rule. As is shown in Table 5.11, the correlations between measured and (average) predicted average wire lengths obtained with the β -models are lower than those obtained with the Rent characteristic, and they decrease as less partitioning levels are used for fitting. However, they are significantly higher than those obtained by using Rent's rule.

Still, for an abstraction of cut probability models to be possible, the relationship between higher-level properties (such as algorithmic complexity or module functionality) and the resulting interconnect complexity must first be made. Similarly, it can be expected that the algorithms and optimization strategies used in synthesis tools have an impact on the final interconnect properties.

5.5 Chapter summary

In this chapter, we have derived a flexible model that allows the prediction of placement wire length distributions for individual circuits and for a wide range of architectural variants. When compared to the results obtained by different grid-based placement tools, the average wire lengths obtained by our model show correlations that are comparable to the correlations between the results of those tools.

When compared to Donath's model, the most important improvement in our own model is caused by using the partitioning Rent characteristic instead of Rent's rule. The flexibility with respect to the architecture model was obtained by relaxing Donath's rigid four-way partitioning framework to a recursive bipartitioning framework that allows slightly unbalanced partitionings.

Since our model is based on a hierarchical placement approach, it is best suited to predict the results of placement tools that use such an approach, at least to obtain an initial placement. However, even with the results of SA, an entirely *flat* placement process, correlations are relatively good.

When Verplaetse's occupation probability function is included in our model, the resulting average wire lengths are within 10% of the values obtained by `Plato` for all but one of the benchmark circuits used. This prediction quality is maintained across a wide range of architectural variants.

Finally, our model was found to be relatively insensitive to variations in the measured Rent characteristic caused by the randomized nature of the partitioning tool. When the β -model is used instead of the Rent characteristic, significant deviations occur for some benchmarks. Additionally, the stability of the predictions decreases as less partitioning levels are used for fitting the β parameter. Still, even when very few partitioning levels are used, the results are better than when Rent's rule is used.

Remember that both Verplaetse's and our own cut probability models are only first attempts, based on empirical observations. Future work aimed at identifying the underlying causes of the observed cut probabilities will hopefully improve those models and lead to even better predictions.

Chapter 6

Probabilistic prediction of the minimal clock cycle

In this chapter we tackle the prediction of the minimal clock cycle, which is one of the most important cost metrics for synchronous digital chips. It is also one of the most difficult metrics to predict.

Within the general philosophy of a priori interconnect prediction, we try to extract those parameters from the circuit interconnect topology that have the biggest impact on the achievable clock cycle. These are used in a new probabilistic model that explicitly takes into account that the minimal clock cycle is determined by the maximum across a large number of combinational path delays.

6.1 The probabilistic longest path problem

Clock cycle prediction is one of the most challenging applications of SLIP techniques. It requires a cascade of models to be applied, the most important of which are shown in Figure 6.1. Since every step represents an approximation, care must be taken to maintain some accuracy across this flow. An additional difficulty is the concept of minimum clock cycle in itself, which is hard to model because it corresponds to the *maximal* combinational path delay.

To explain the nature of the longest path problem, let's return to the simple sequential circuit example from Chapter 2 (Figure 6.2). The essence of sequential circuit design is that signal transitions are *kept in pace* by a global clock signal. The flipflops or other clocked gates in the

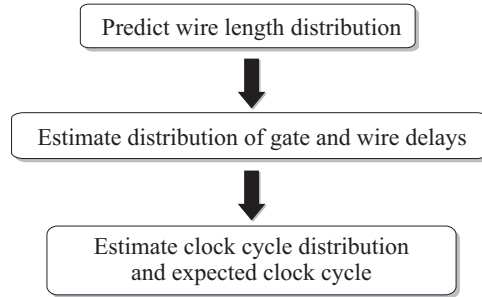


Figure 6.1: The main stages in the estimation flow for clock cycle prediction.

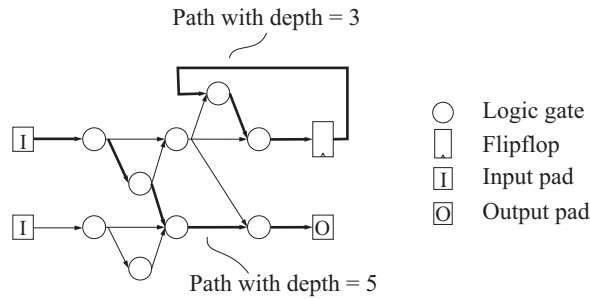


Figure 6.2: Schematic representation of a simple circuit graph, indicating flipflops, input pads, output pads, logic gates and directed nets.

circuit pass on the signal values at their inputs to their outputs only at times indicated by the clock signal, for instance at the clock signal's rising edge. Until the next rising edge on the clock signal, the flipflop “remembers” that value, and its output signal is no longer affected by transitions on its input signal. The signal values at the flipflop outputs then propagate through the combinational gates and interconnect.

In a sequential circuit graph, directed sequences of blocks connected by nets can be identified. Such a sequence is called a *combinational path* (called *path* hereafter), if it connects a sequence of logic gates only, and if it starts at a flipflop or an input pad and ends at a flipflop input or an output pad.¹ In well-designed synchronous circuits, such paths must never contain cycles. In the circuit graph of Figure 6.2, several such paths are shown. We will refer to the combination of a source-to-sink connection and its driving block as a *path segment*. The number of segments on a path is called the path's *logic depth* (or simply the path

¹We assume that both input signals and output signals are clocked.

depth). After the circuit is implemented, each segment has a certain *delay* between transitions on its input signals and the associated transitions on its output signals, consisting in part of the gate delay and in part of the interconnect delay. The sum of all segment delays on a combinational path is the path delay.

To ensure a correct functionality, all flipflop input signals must have reached their final value and be stable sufficiently long before the next rising edge on the clock signal (the *setup time*) and remain stable for some time after this event (the *hold time*). Hence, the minimal clock cycle that still ensures a correct functionality is determined by the longest delay occurring on any combinational path in the circuit (the *maximal combinational path delay*).

As discussed in Chapter 1 the segment delays have long been dominated by the gate delays. Hence, the minimal clock cycle was mainly determined by the maximal logic depth occurring in the circuit. However, due to the technology scaling trends explained in Section 1.2, the longest interconnect delays are now comparable to or exceeding some of the gate delays. Therefore, paths with less than maximal logic depth can end up having the longest delay. The difficulty in clock cycle prediction is that the actual segment delays are not known before the circuit is implemented. Instead, in the a priori prediction framework adopted in this Ph.D. thesis, they have a probabilistic nature. As a result, the minimal clock cycle estimate also becomes probabilistic.

Most current techniques for the a priori prediction of the minimal clock cycle are still based on the maximal logic depth and the average wire length. The gate and interconnect delay corresponding to a wire of average length are computed (using, e.g., [Sak93]) and the result is multiplied by the maximum logic depth that occurs in the design to model the critical path delay. In some cases, a distinction is made between wires with different fanouts, and often one average wire is replaced by a wire of maximal length, to model the possibility of a very long (global) connection occurring in the critical path. Adding the setup and hold times and the clock skew to this result leads to an estimation of the minimal cycle time.

In [ISHC02], Iqbal et al. offered an improvement by first calculating the wire delay distribution from the wire length distribution², and then stochastically sampling this distribution to find the minimal clock

²For calculating this delay distribution, they included routing layer assignment and different technology parameters for each layer in their model.

cycle. Since the wire delay is a non-linear function of the wire length, its average is not equal to the delay of a wire of average length. Hence, they have effectively improved the second stage in the estimation cascade of Figure 6.1. Furthermore, the fact that they offer a distribution for the clock cycle reflects the fact that the design process is inherently stochastic.

However, none of the previous techniques models the fact that the clock cycle is determined by the maximum delay across a large number of paths. Particularly, in circuits that are strongly pipelined, we can expect that there exist many paths with maximum logic depth, but only the slowest one really matters. Furthermore, there are many other paths in the circuit with depths just below the maximal logic depth that might also turn out to be the slowest after implementation.

In this chapter, we focus on improving the final step in the cascade of Figure 6.1. As in [ISHC02], we offer a probabilistic view on minimal clock cycle. We use two main probabilistic concepts: the distribution of the sum of a number of independent random variables, and the distribution of the largest of a number of such variables (see Appendix B). Combining these two and taking the parallelism in the circuit topology into account, we derive a model for the clock cycle distribution that corresponds better with the experimental distribution, measured across a large number of circuit implementations. The principal results presented in this chapter were published in [DSVC02].

Recently, this problem has received considerable attention in the context of yield prediction [Sch02, ABZ⁺02, ABZV02, KO02, AYNM02]. With respect to clock cycle, the yield is the fraction of fabricated chips that meet the clock cycle constraint. To ensure that the yield is sufficiently high, a tighter constraint is targeted during design than the one that must actually be met. This allows for some delay fluctuations due to, e.g., process variation or delay drift caused by heat effects. However, whereas the targeted clock cycles are ever decreasing, the physical phenomena that cause delay variations remain largely the same. Hence, their relative importance rapidly increases. From the designers' point of view, the main worry is that it has become very difficult to achieve timing closure. Therefore, they do not want to decrease the clock cycle constraints more than is strictly necessary to achieve the targeted yield. Hence, a first problem they want addressed is the prediction of the minimal margin on the clock cycle that has to be allowed to ensure that the desired yield can be achieved.

Second, a validation of this yield must still be performed after physical design and before fabrication. At this stage, nominal delays for all gates and interconnect can be extracted. Together, they ensure that timing closure is just met with respect to the design clock cycle constraint. However, at this stage, the exact delay fluctuations are not yet known, so accurate probabilistic models are required to predict the distribution of the actual clock cycle across the chips after fabrication.

While the problem formulation of yield prediction with respect to the clock cycle is the same as the problem of predicting the clock cycle before physical design, the boundary conditions are quite different. For yield prediction, the delay fluctuations are typically assumed to follow a normal distribution, with relatively small standard deviations (a few percent of the nominal values). The required accuracy is very high, but computation time is not really an issue, as long as it remains within reasonable limits. In contrast, the delay variations we need to consider are derived from the wire length distributions, which are far from normal and have standard deviations of up to several times their mean. Because of the limited accuracy that can be achieved in an a priori prediction flow, acceptable error margins are much larger, but computation time is definitely an issue.

6.2 The segment delay distribution

What we aim to predict is the distribution of the clock cycle across all implementations of a circuit with a given topological (graph) representation, produced by a given randomized implementation process \mathcal{P} . In this context we define the *segment delay distribution* $P(d)$ as the probability mass function for the delay of a randomly selected segment in a randomly selected circuit implementation resulting from \mathcal{P} . Depending on the models used to derive this distribution, it can include stochastic variation caused by process variation [Ber97] as well as optimal gate sizing and buffer insertion [CP99, JB00, HO00]. Note that the derivation and evaluation of such models (i.e., the central box in Figure 6.1) is not within the scope of this Ph.D. thesis. The input to our clock cycle prediction technique is a segment delay distribution, without any restrictions as to how it was derived. Hence, to validate our model, we only use a subset of the broad range of wire length related delay models that are currently available.

To estimate segment delays, we have applied the delay models

	2001	2004	2007	2010	2013	2016
Physical gate length (nm)	65	37	25	18	13	9
Wiring pitch (nm)	350	210	150	105	75	50
Wiring aspect ratio	1.6	1.7	1.7	1.8	1.9	2.0
ρ ($\mu\Omega - cm$)	2.2	2.2	2.2	2.2	2.2	2.2
T_{ox} (nm)	2.3	2.0	1.4	1.2	1.0	0.9
V_{th} (V)	0.19	0.12	0.05	0.021	0.003	0.003
V_{dd} (V)	1.2	1.0	0.7	0.6	0.5	0.4
I_{dsat} ($\mu A/\mu m$)	900	900	900	1200	1500	1500

Table 6.1: Technology parameters taken from the ITRS, 2001 edition (MPU data)

used in BACPAC [SK99] (models, amongst others, from [SK99, Sak93, CHA⁺92]). The technological parameters used in those models were taken from the 2001 edition of the ITRS [ITR01]. Since clock cycle prediction models are often used in technology exploration, we have performed our calculations for some of the technology generations from 2001 onwards (Table 6.1). Note that we have used technology parameters corresponding to local interconnect for all wire lengths and no buffering or gate sizing was considered. However, all of these aspects can be included when translating the wire length distribution to the segment delay distribution by using approximations from, e.g., [ISHC02] (layer assignment) and [SK99] (optimal area effective buffer insertion). We are aware that our choice of delay models is not state-of-the-art and the results presented in this chapter are not intended to study clock cycle scaling trends across the technology generations. They only serve to demonstrate the performance of our clock cycle prediction model as the impact of interconnect delay increases.

6.3 Probabilistic clock cycle prediction

After a delay estimation is made for every path segment, the path delay for each path can be estimated by the sum of its segment delays. The purpose of clock cycle analysis is to find the maximal combinational path delay that occurs in the circuit. In this section, we propose a model to estimate the probability mass function of this maximal delay across many circuit implementations, using the segment delay distribution and parameters extracted from the circuit topology. Our model is

based on two types of derived distributions: the distribution of the sum of a number of segment delays and the distribution of the maximum of a number of path delays.

6.3.1 Some probability theory

If d_1 and d_2 are two independent discrete probabilistic segment delays³ with probability mass functions (pmfs) $P(d_1 = c_i)$ and $P(d_2 = c_i)$ for any value c_i , the pmf of their sum $d = d_1 + d_2$ is given by (see Appendix B):

$$P(d = \delta) = \sum_{i=0}^{\delta} P(d_1 = c_i)P(d_2 = \delta - c_i) \quad (6.1)$$

where c_i is the i -th delay value. Similarly, the delay distribution for a sequence of n statistically independent segments can be calculated as the n -fold discrete convolution of the individual segment delay distributions (which are not necessarily equal). For calculating the distribution of n -stage path delays, this convolution approach is equivalent to the Monte Carlo sampling approach reported in [ISHC02]. They estimated the delay for a single path of maximum logic depth by randomly taking n independent samples from the segment delay distribution and then adding them to yield the path delay. The path delay distribution was obtained by repeating this process a very large number of times.

Based on the path delay distribution, the distribution of the maximal path delay across m *statistically independent* paths can be computed from their respective cumulative path delay distributions. The cumulative distribution $P(d \leq \delta)$ of the maximum across m independent path delays d_1, d_2, \dots, d_m is given by:

$$P(\max\{d_1, d_2, \dots, d_m\} \leq \delta) = \prod_{i=1}^m P(d_i \leq \delta). \quad (6.2)$$

If the path delays are equally distributed, this expression reduces to:

$$P(\max\{d_1, d_2, \dots, d_m\} \leq \delta) = P(d \leq \delta)^m. \quad (6.3)$$

The pmf of the maximum path delay can then be calculated from the resulting cumulative distribution.

³We will always consider the segment delays to be discretized into equally spaced “buckets” with central values c_i

6.3.2 Equivalent independent paths

In a graph that consists of independent (i.e., disjoint) paths only, and if the number of independent paths is known for every occurring path depth, Equations (6.1) and (6.2) or (6.3) can be used to estimate the maximal path delay distribution for each depth. Equation (6.2) results in the maximal delay distribution across all path depths.

However, the paths occurring in real circuits are far from independent. For small circuit graphs, it is possible to model the entire complex network of dependencies of path delays on segment delays [OK02]. However, this approach is no longer feasible in a graph that consists of hundreds of thousands or even over a million nets.

For that reason, we wish to estimate a *delay-equivalent graph topology* for the circuit graph, i.e., a topology that would lead to approximately the same maximal path delay distribution, but for which the paths of each depth are fully independent. To do this, we define the notion of *segment criticality*. Consider a segment that occurs in multiple paths. These paths are obviously not independent. Of all paths to which the segment belongs, we take the one(s) with the longest depth. Intuitively, the segment's delay will have the biggest impact on these paths of largest depth. We define this depth as the segment's criticality.

Our method then rests on two simplifications. The first simplification is to disregard the impact of a segment's delay on any path with smaller depth than its criticality. A second simplification is the assumption that all segments with a given criticality contribute equally to the maximal path delay for the corresponding depth and that they form independent paths. With these simplifications, we can estimate the *number of equivalent independent paths (equipaths)* for each path depth, i.e., the number of independent paths of that depth in the delay equivalent graph topology. We proceed as follows. First we determine the segment criticalities. We consider the set of segments with a given criticality. Using this set, we construct a set of independent "equivalent paths", by simply dividing the number of segments in the set by the path depth (their common criticality). We then apply Equation 6.3 to this set of "paths". Since the division often results in non-integer numbers of "paths", we need some kind of interpolation scheme. The easiest way to provide this is to replace the integer power in Equation (6.3) by a non-integer one. Note that our equivalent graph topology consists of the same number of segments as the original graph, although the number of paths is lower.

Depth \mathcal{D}	Number of paths	Number of segments	Criticality frequency	Independent paths
1	0	0	0	0
2	0	0	0	0
3	1	3	1	0.33
4	3	9	1	0.25
5	5	15	9	1.8
6	1	6	6	1

Table 6.2: Path depth analysis for the simple circuit of Figure 6.2

Though our approach is obviously only a rough estimation, based on mainly intuitive principles, it turns out to capture the essence of the parallelism, present in many circuit topologies. Remark that this method is akin to the notion of 'degrees of freedom' (d.o.f.) in common statistical techniques. These are found by counting all individual random variables (here: the delays of *all* paths) and afterwards reducing this number by the number of relationships that exist among them. It is well known that the values of, e.g., test percentiles strongly depend on the d.o.f.-number.

6.3.3 Computing the segment criticalities

Examining the example circuit of Figure 6.2, we find that the maximal depth is six and that there is only one path with that depth. Also, there are five paths of depth five, but they all share segments with each other and with the path of maximal depth. Among them, these five paths use fifteen segments, only nine of which are not part of a path of depth six. Hence, there are six segments with criticality six and nine with criticality five. The same analysis can be performed for all path depths, leading to the results in Table 6.2. The second and third columns in this table show the total number of paths for each depth and the total number of segments on these paths, respectively. The fourth column shows the frequency distribution of the criticalities across all segments.

For large circuit graphs, performing such an analysis would be infeasible because of the combinatorial "explosion" of the total number of paths. However, it is possible to compute the segment criticalities in approximately linear time by traversing the circuit graph in the same way this would be done to determine the minimal and maximal path delays after layout (see Figure 6.3). In delay analysis, the processing of

```

function TraverseGraph()
{
  Mark Initial blocks (Input pads and flipflops);
  While unmarked blocks
  {
    CandidateBlocks={};
    for each unmarked block Bi
    {
      If(all Predecessors(Bi) marked)
        CandidateBlocks={CandidateBlocks,Bi};
    }
    Process(CandidateBlocks);
    Mark(CandidateBlocks);
  }
  ExtractResults();
}

```

Figure 6.3: Graph-traversal algorithm for determining the terminal criticalities

a block circuit element or B_i consists of determining the maximal (and minimal) delay to reach that block. In our algorithm, it consists of creating a *local criticality list* for that block (Figure 6.4). This list contains all path segments that have been traversed on some path to reach that block. For each of these segments, we keep the maximal path depth (local criticality) occurring on a path to B_i that runs through this segment.

The local criticality list for a block can be created from the local criticality lists of its predecessor blocks. A list entry is created for all path segments that occur in any of the predecessor's lists. The local criticality for that segment in the current block is the maximum amongst its criticalities in the predecessor's lists, incremented by one. For instance, in Figure 6.4, we find for the local criticality $crit(c, s_1)$ of path segment s_1 at block c :

$$crit(c, s_1) = \max(crit(a, s_1), crit(b, s_1)) + 1 = 3 \quad (6.4)$$

To save memory, a block's local criticality list can be deleted as soon as all of its successor blocks have been processed. When all blocks have been processed, only the criticality lists for flipflops and output pads remain. These are then combined into the global segment criticality

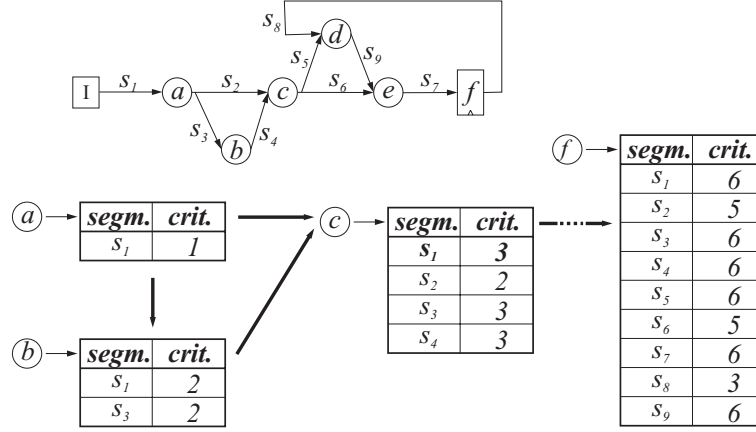


Figure 6.4: Evolution of local criticality lists for a small subcircuit.

distribution, by taking the maximal criticality that occurs for each segment.

6.4 Model validation

To validate our model, and compare it to the traditional approach (average delay of a single path with maximal depth, with equally distributed segment delays), we have performed a large number of experiments on benchmarks from the LGSynth benchmark series [CBL]. Since some of these benchmarks are very small, we have used only the ones with more than 500 gates. This leaves a total of 68 benchmarks, with sizes ranging from 527 to 24819 blocks and maximal logic depths ranging from 6 to 284. As was the case for all other benchmarks used in this work, all multi-terminal nets were converted to multiple two-terminal nets.

For each of these benchmarks, we have performed 100 placement experiments with `Plato`.⁴ The resulting wire lengths were used to estimate the segment delays as mentioned in Section 6.2. We have only computed the distribution of the maximal path delay. Hence, setup and hold times were not included. We also did not incorporate clock skew or noise. Note that these can easily be added to the model, but

⁴Note that in these experiments, no explicit clock cycle optimization was performed. As in the previous chapter, the placement cost function was total wire length.

we did not find predictions of those intervals for the future technology generations mentioned in the 2001 edition of the ITRS. The result of our experiments is a maximal path delay distribution (as well as the average maximal path delay) for each benchmark circuit.

For our clock prediction model, the wire length distribution for each benchmark was first averaged across the 100 placement experiments. Then, using the same models as for the experiments, this distribution was converted to a discrete segment delay distribution. The equipaths were determined once for each circuit topology.

The main goal of our new model is to be able to predict the expected maximal path delay. Figure 6.5 shows the histogram of the relative errors on this delay (compared to the experimental average) of the traditional prediction and our new prediction for the ITRS technological parameters (technology node 2001, all benchmarks). It is clear that for most benchmarks, the traditional approach results in a significant under-estimation of the average maximal path delay. Furthermore, the correlation that is achieved with the traditional model (its reliability) decreases as the impact of interconnect gets larger. In contrast, our model, on average, leads to much better results, and its reliability is maintained across all technology generations.

This increased accuracy is maintained across all technology generations mentioned in Table 6.1 as can be seen from Table 6.3.

However, for some benchmarks, our model still results in either a large under-estimation, or a large over-estimation. The worst over-estimation occurs for benchmark circuit `dsip`, while the worst under-estimation occurs for benchmark circuit `rot`. Figures 6.6 (a) and (b) show the experimental and predicted probability distribution for the maximal path delay. Note that even the traditional model results in an over-estimation for benchmark `dsip`. These outliers are probably due to the homogeneity assumption, that forms the basis of both our model and the traditional model. For some benchmarks, however, there is a systematic deviation from this assumption for some segments. Their delays can be systematically higher than the average segment delay if they are a part of large-fanout nets or if they are in a subcircuit with high local interconnect complexity. Segments can have a systematically smaller delay if they are located in a part of the circuit with low local interconnect complexity. If a significant number of the segments on paths with large depth have either a systematically higher or a systematically lower delay, this must reflect on the global maximal path delay.

Traditional prediction						
	2001	2004	2007	2010	2013	2016
Correlation	0.954	0.952	0.951	0.951	0.949	0.948
Avg. error (%)	-24.02	-25.86	-25.76	-25.78	-26.18	-26.78
Min. error (%)	-57.12	-58.11	-58.12	-58.25	-58.62	-59.14
Max. error (%)	30.04	29.50	29.55	29.62	29.61	29.54
% in [-10%,10%]	23.08	18.46	16.92	16.92	16.92	16.92
% in [-20%,20%]	38.46	36.92	36.92	36.92	36.92	36.92

New prediction						
	2001	2004	2007	2010	2013	2016
Correlation	0.972	0.971	0.971	0.971	0.971	0.971
Avg. error (%)	9.85	8.94	8.98	8.95	8.74	8.43
Min. error (%)	-33.40	-35.03	-34.90	-34.81	-35.04	-35.46
Max. error (%)	90.92	91.79	91.66	91.48	91.46	91.54
% in [-10%,10%]	58.46	60.00	60.00	60.00	60.00	56.92
% in [-20%,20%]	75.38	76.92	76.92	76.92	76.92	78.46

Table 6.3: Performance of traditional prediction and new prediction across technology generations and across all benchmarks, compared to experimental results: correlation, average relative error, smallest and largest occurring relative errors and percentage of benchmarks for which the prediction is within 10% and 20%, respectively.

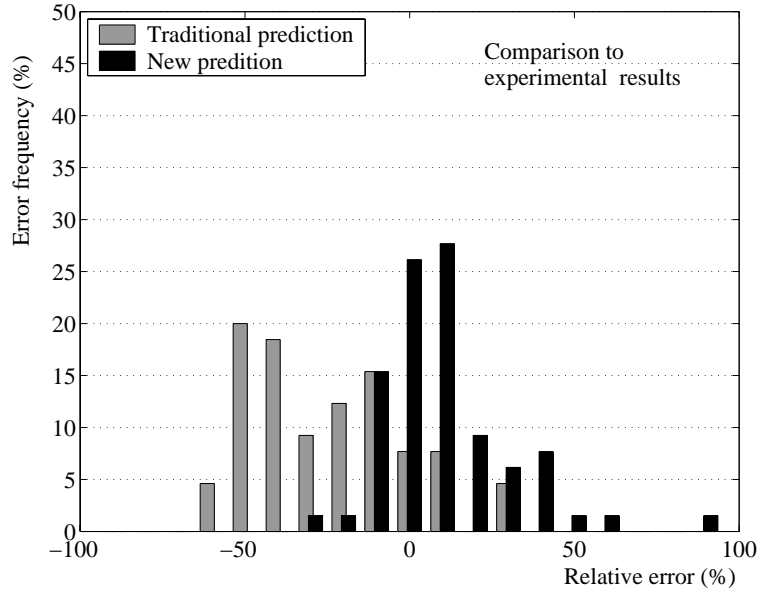


Figure 6.5: Histogram of relative errors of average maximal path delays predicted by the traditional and our new probabilistic model for the 2001 technology node (compared to experimental results).

To illustrate the impact of the homogeneity assumption, we have performed Monte Carlo simulations of the maximal path delay. In each iteration, the segment delays for all segments in the circuit graph were randomly sampled from the same global segment delay distribution that was used for the predictions. Then, for that assignment of segment delays, the maximal path delay for the was calculated as if they were the result of a real placement. In each experiment, 1000 such iterations were performed. The setup of these Monte Carlo experiments ensures that the homogeneity assumption is valid. The resulting maximal path delay distributions for benchmarks `rot` and `dsip` are also shown in Figures 6.6 (a) and (b). The distribution of the relative deviations of both models to the average maximal path delays of the Monte Carlo experiments are shown in Figure 6.7, while Table 6.4 shows the evolution of these results across the ITRS technology generations.

As could be expected, we find that our predictions agree better with the Monte Carlo results than with the results obtained from real placements. The correlation of our prediction model with the Monte Carlo results is almost perfect, which shows that we have indeed succeeded

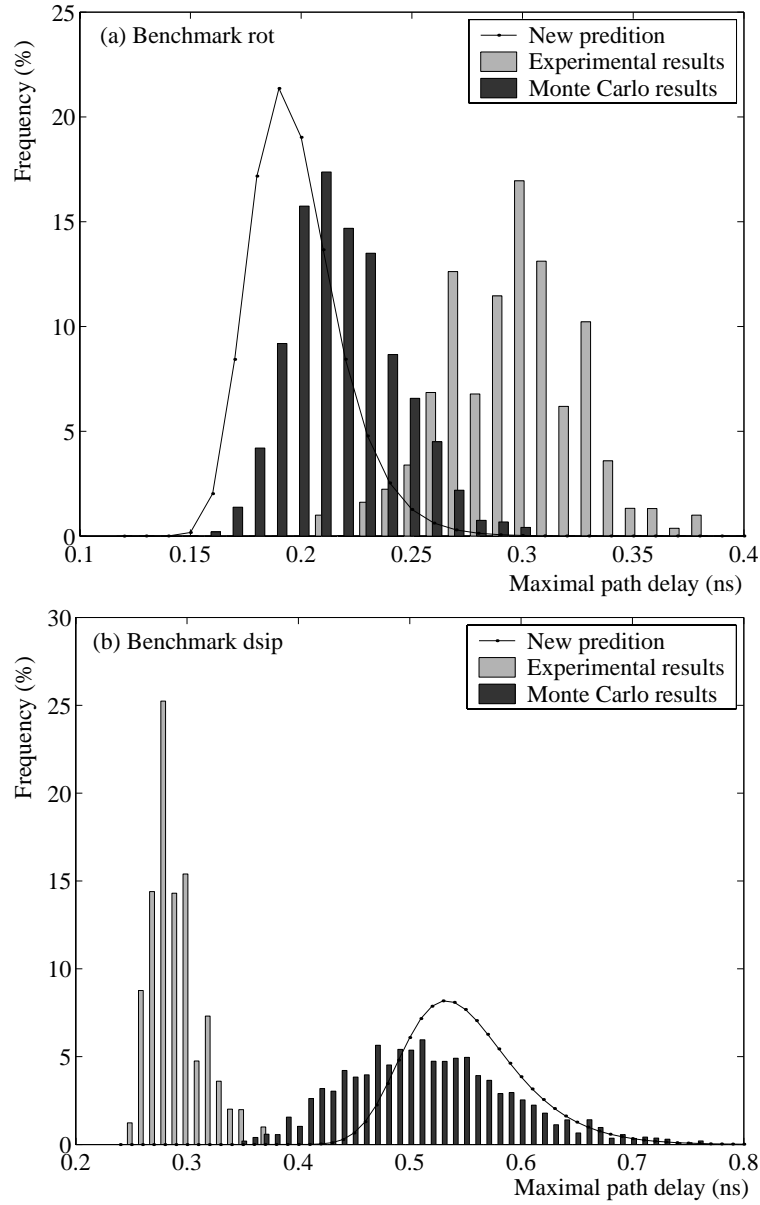


Figure 6.6: Experimental and Monte Carlo relative maximal path delay frequencies, compared to prediction with our new model for: (a) *rot*, the benchmark for which our model yields the worst under-estimation, and (b) *dsip*, the benchmark for which our model yields the worst over-estimation (2001 technology node).

Traditional prediction						
	2001	2004	2007	2010	2013	2016
Correlation	0.981	0.982	0.981	0.981	0.981	0.981
Avg. error (%)	-34.55	-36.39	-36.27	-36.24	-36.59	-37.14
Min. error (%)	-62.49	-63.38	-63.37	-63.46	-63.74	-64.11
Max. error (%)	6.75	4.76	4.94	5.11	4.92	4.48
% in [-10%,10%]	3.08	3.08	3.08	3.08	3.08	3.08
% in [-20%,20%]	20.00	12.31	13.85	13.85	12.31	12.31

New prediction						
	2001	2004	2007	2010	2013	2016
Correlation	0.997	0.996	0.996	0.996	0.996	0.996
Avg. error (%)	-4.62	-5.73	-5.65	-5.60	-5.77	-6.07
Min. error (%)	-18.88	-20.53	-20.42	-20.38	-20.71	-21.24
Max. error (%)	18.49	18.21	18.22	18.19	18.10	17.98
% in [-10%,10%]	80.00	69.23	69.23	69.23	69.23	69.23
% in [-20%,20%]	100.00	98.46	98.46	98.46	96.92	96.92

Table 6.4: Performance of traditional prediction and new prediction across technology generations and across all benchmarks, comparison to Monte Carlo results.

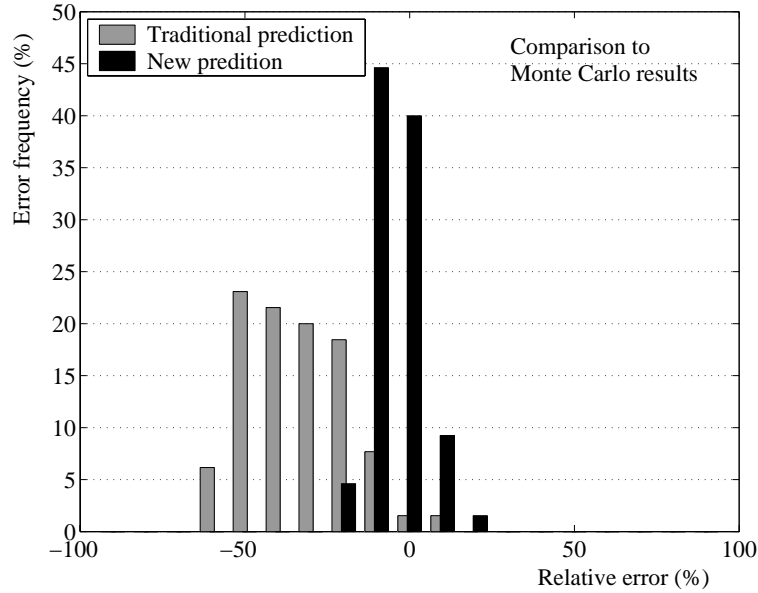


Figure 6.7: Histogram of relative deviations from Monte Carlo results of the average maximal path delays predicted by the traditional and our new probabilistic model (2001 technology node).

in modeling the essential impact of parallelism on maximal path delay. Note that the traditional approach (using a single path of maximal depth) also implicitly assumes that the segment delays are equally distributed (the homogeneity assumption). Therefore, it also correlates better with the Monte Carlo results than with the real placement results. However, this correlation remains a lot lower than that obtained with our new model.

Still, compared to the Monte Carlo results, our model tends to slightly underestimate the average maximal path delays by (on average) between five and eight percent. This is probably due to the fact that our equivalent graph topology consists of fully independent paths. However, while maintaining the same number of segments, other topologies exist for which the maximal path delay distribution can still be calculated. For instance, consider the two topologies of Figure 6.8. Both consist of the same number of segments, but there are five blocks and four paths in topology (2) while there are six blocks and only two paths in topology (1). If the segments s_1 , s_2 , s_3 and s_4 have delays d_1 , d_2 , d_3 and d_4 , respectively, the maximal path delay of

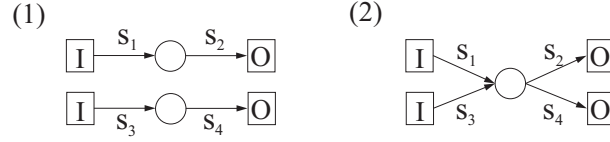


Figure 6.8: Two graph topologies with equal number of segments, but with different maximal path delay distributions.

topology (1) is equal to $\max(d_1 + d_2, d_3 + d_4)$, while that of topology (2) equals $\max(d_1, d_3) + \max(d_2, d_4)$. The expected average maximal path delay of the second topology is larger than that of the first. Indeed, using the segment delay distribution of benchmark `dsip` with the 2001 technology parameters, we find that the average maximal path delay of the first topology is almost 7% below that of the second (Figure 6.9).

In general, as soon as some interaction (dependence) between paths occurs in the real path topology for a given path depth, the maximal path delay for that depth will be greater than that caused by fully independent paths with the same number of segments. As this is generally the case in real circuits, even better results might be obtained by creating equivalent path topologies that not only maintain the number of segments, but also the number of distinct blocks.

6.5 Using predicted wire length distributions

In the previous section, we have evaluated our new approach to the last stage in the prediction flow of Figure 6.1 by using the measured wire length distribution. In this section, to complete the clock-cycle prediction flow, we combine the last stage with the predicted wire length distribution from the previous chapter.

From Table 6.5, we can see that the quality of our prediction models, when applied to the `LGSynth` benchmarks, is similar to the results obtained in the previous chapter.

Figure 6.10 shows the correlations and average relative errors of clock cycle predictions compared to placement results and the Monte Carlo results. The predictions were obtained with the traditional model and our new model, once using the measured wire length distributions and once using predicted wire length distributions from our full prediction model and with the three different occupation probabilities. We

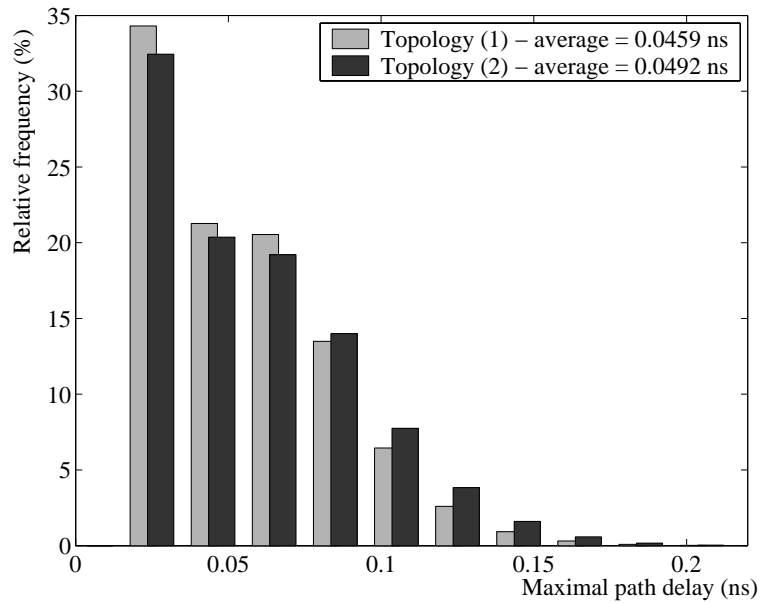


Figure 6.9: Maximal path delay distributions for the two graph topologies shown in Figure 6.8, calculated from the segment delay distribution of benchmark `dsip`, using the 2001 technology parameters.

	Full model results	
	without occ. prob.	with Verplaetse's occ. prob.
Avg. error (%)	36.84	4.71
Min. error (%)	27.12	-7.10
Max. error (%)	48.22	16.51
Correlation	0.998	0.996

Table 6.5: Average, minimal and maximal relative errors on the predicted average wire lengths (using our full prediction model without occupation probability and with Verplaetse's occupation probability) across all 65 benchmarks used in this chapter. The correlations between measured and predicted average wire lengths are also shown.

New prediction with predicted wire length distribution (Verplaetse's occupation probability)						
	2001	2004	2007	2010	2013	2016
Correlation	0.967	0.966	0.966	0.966	0.966	0.966
Avg. error (%)	14.45	13.85	13.90	13.96	13.91	13.78
Min. error (%)	-32.96	-34.39	-34.26	-34.15	-34.33	-34.67
Max. error (%)	101.93	103.28	103.26	103.38	103.82	104.35
% in [-10%,10%]	44.62	46.15	46.15	46.15	46.15	46.15
% in [-20%,20%]	70.77	70.77	70.77	70.77	70.77	72.31

Table 6.6: Performance of our new model for predicting clock cycle distributions when used in combination with predicted wire length distributions (Verplaetse's occupation probability).

can conclude from this figure that the accuracy of the predicted wire length distribution hardly affects the correlations between measured and predicted clock cycles. There is an almost constant loss of reliability, compared to predictions that use the measured wire length distributions. The obtained correlations remain significantly better than those obtained with the traditional approach to clock cycle prediction. Obviously, the quality of the predicted wire length distribution does affect the accuracy of the predicted clock cycle. Because a maximum is taken, even the small over-estimation that is present in our model with Verplaetse's occupation probability results in a significant increase in the predicted clock cycles. This can also be concluded from the resulting error distributions shown in Figure 6.11.

To conclude, Table 6.6 summarizes the results when using our full prediction model with Verplaetse's occupation probability.

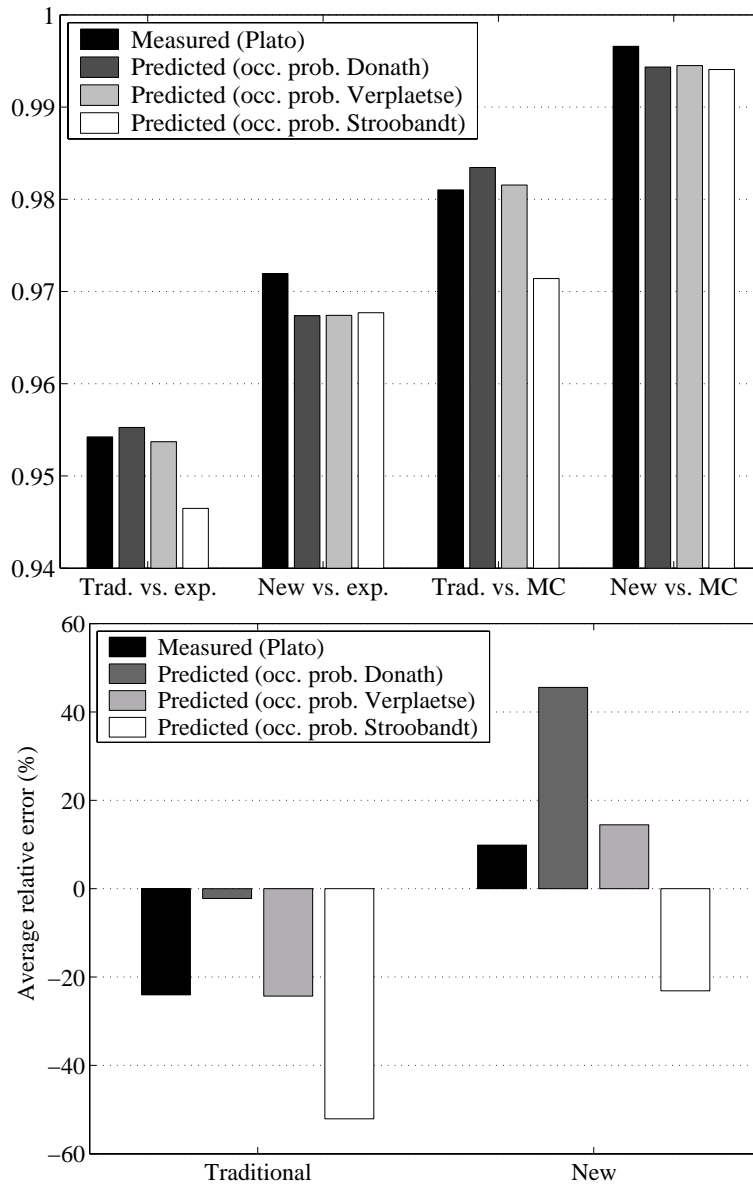


Figure 6.10: Correlation values and average relative errors obtained when measured (Plato) and predicted wire length distributions are used in the traditional prediction and our new prediction. All predicted distributions were obtained with our full model, using Donath's uniform occupation probability, Verplaetse's occupation probability and Stroobandt's occupation probability, respectively (technology node 2001). For the correlations, the comparison with the Monte Carlo results is also shown.

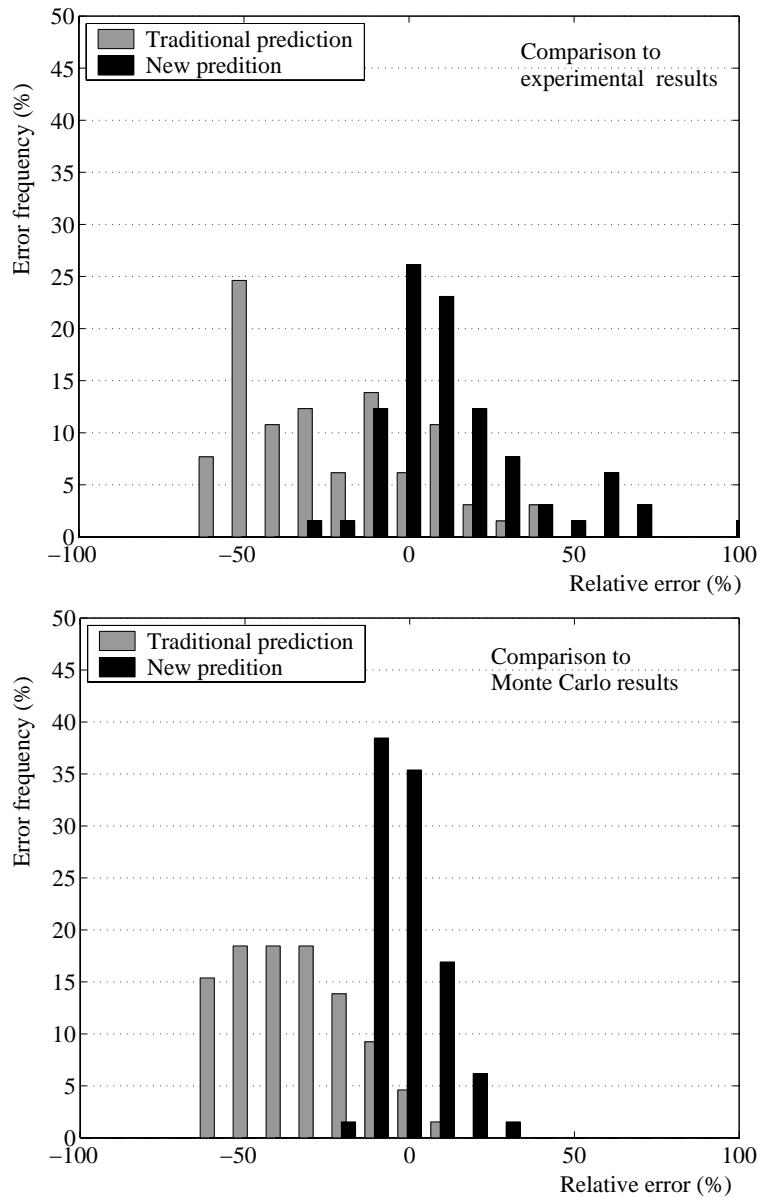


Figure 6.11: Histogram of relative errors of average maximal path delays predicted by the traditional and our new probabilistic model, using predicted wire length distributions (with Verplaetse's occupation probability) for the 2001 technology node. The top graph shows the comparison with the `Plato` experimental results, while the bottom graph shows the comparison with the Monte Carlo results.

6.6 Conclusions

In this chapter, we have presented a new probabilistic model for predicting the maximal path delay of synchronous circuits, for placements that optimize total wire length. Our model is based on the segment delay distribution, the circuit topology and the homogeneity assumption. Experimental results show that, by modeling the clock cycle as a maximum across a number of path delays, our model correlates a lot better with experimental results than the traditional approach, where the maximal path delay is predicted as the average delay of a single path. The absolute accuracy, expressed by the average relative prediction error, also increases. Although it is very simple, we can conclude that the proposed model succeeds in capturing most of the impact of the parallelism that is present in real circuits.

We have shown by Monte Carlo simulation that most of the largest prediction errors are due the homogeneity assumption, which is not only used in our model, but also in the traditional approach. It appears that the cumulative effect of local deviations to that assumption has a large impact on the maximal path delay for some benchmarks.

Additionally, we have indicated a probable cause for the remaining (systematic) prediction errors in our model. These are probably due to our equivalent path topologies and to the fact that, while maintaining the number of segments, they do not maintain the number of blocks.

Finally, we have shown that, when combined with predicted wire length distributions, the overall reliability of the predicted maximal path delays, expressed by the correlation between measured and predicted results, does not degrade much. However, as is illustrated by the results obtained without occupation probability, accurate clock cycle predictions are only achievable if all steps in the prediction chain are sufficiently accurate.

Chapter 7

Directions for further research

In this chapter, we summarize the main topics for further research and offer some possible starting points for their solution. The first section addresses the issues that are directly related to extending the interconnect prediction models we have developed in Chapters 5 and 6. We also discuss the most urgent extensions to the underlying framework of assumptions.

In the second section of this chapter, we offer some suggestions on how to get more insight into the fundamental relationship between the interconnect topology of the circuit graph and the achievable level of placement optimization. To a large extent, this discussion focuses on the identification of promising related work from other research domains.

7.1 Upwards: further model refinement and extensions

Although we have achieved significant improvements to the reliability and accuracy of a priori interconnect prediction techniques, a lot of issues remain to be resolved. Some of those can be addressed within the assumptions adopted in this work: the conversion of multi-terminal nets to two-terminal nets, the homogeneity assumption and the use of placement wire lengths as an estimate for wire lengths resulting from a complete physical design flow. However, the most urgent topics for further research target the validation and, where necessary, the relaxation of those assumptions.

7.1.1 Direct extensions of this work

Prediction of the wire length distribution

Within the framework of assumptions adopted in this Ph.D. thesis, predictions can probably be made even more accurate, including the level of optimization reached by particular tool flows.

In Chapter 5, we have shown that Verplaetse's occupation probability function is a good candidate for modeling the optimization achieved by performing optimal module swapping in a partitioning-based placement approach. Essentially, it is a combination of the theoretically derived occupation probability proposed by Stroobandt and a *cutoff length*, below which the level of optimization rapidly decreases. This concept accords well with what happens during incremental placement optimization, even for a flat placement approach. Indeed, a gate is usually connected to a number of nets. Moving this gate can improve the length of some connected nets, but it will usually increase the length of others. The shorter the nets connected to a particular gate, the lower the probability that moving it will result in an overall length decrease. Hence, it is easier to optimize long nets than short ones.

It can be hoped that the approach of introducing a parameterized cutoff length in the occupation probability function will also help building parameterized models for tool-specific optimization. These can be a mix of hierarchical occupation probabilities (i.e., depending on the module dimensions) and a global one that is independent of the module dimensions.

Also, while we have moved away from Rent's rule in our extended partitioning-based prediction model, it is still used in the expressions for the occupation probabilities. The improvements that were achieved by using the Rent characteristic for calculating the number of cut nets are impressive. If we could also express the occupation probabilities as a function of the Rent characteristic, a further increase of the achieved correlations might be obtained.

Besides the continued exploration of occupation probabilities with different cutoff lengths, we also need to study the impact of other circuit parameters such as the circuit size G_c , the number of nets N_c or the maximal block degree on tool performance.

Another issue that needs further study is the abstraction of the proposed models to higher levels in the design hierarchy. This will cer-

tainly require the use of parameterized terminal gate-relationships such as the recursive partitioning model proposed by Verplaetse or our own adaptation of it. In this context, an empirical study of the relationship between the functionality of circuit modules and the parameter values that are to be used in such models would probably yield important information. A similar study with respect to the impact of R/T-level and logic level synthesis and optimization tools might prove even more important.

Prediction of the minimal clock cycle

The model for clock cycle prediction proposed in Chapter 6 mainly serves as an illustration of the facts that:

- (a) as the relative importance of interconnect delay increases, and because of the large variations on interconnect delay due to interconnect length, it is necessary to model the minimal clock cycle as a *maximum* instead of an average;
- (b) it is relatively easy to establish a model, based on the concept of a delay equivalent graph topology, that captures the essence of the circuit topology with respect to clock cycle.

However, this model is clearly no more than a first attempt. A more thorough analysis of experimental results, as well as the underlying mathematics and statistics are required to obtain increased accuracy. In particular, the impact of local effects, causing some wires to be systematically longer than others, needs to be studied and incorporated into the model.

Additionally, our model does not incorporate the effects of clock cycle optimization during physical design. Obvious extensions are the inclusion of buffer insertion, gate sizing and optimal routing layer assignment in the models used for converting wire length to segment delay. However, clock cycle optimization during placement is probably an even more important factor. For instance, the aim of such optimization is to put more effort into optimizing the most critical nets. As a result, the homogeneity assumption will no longer be fulfilled as segments with higher criticalities will, on average, have shorter wires. However, it can be expected that the level of optimization that can be reached will depend on the criticality distribution of the path segments. As there will always remain long wires in the final placement, the wires

on the most critical paths can only be kept short if there is a sufficiently large number of less critical paths that can afford longer wires. In the extreme case where all paths have equal depth, only very little clock cycle optimization can be achieved during placement. However, to our knowledge, the relationship between the distribution of segment criticalities and the achievable clock cycle (expressed as a function of placement wire lengths) has not yet been studied.

7.1.2 The homogeneity assumption and multi-terminal nets

One of the key assumptions made in this work is the homogeneity assumption, which states that, considered across a large number of placement experiments, the lengths of individual source-to-sink connections are equally distributed random variables. From the results presented in Chapter 6, we can conclude that this assumption does not always hold: some connections have a tendency to end up longer than average, while others have a high probability of being implemented by short wires. While this does not greatly affect the wire length distribution, it does become important when extreme values (i.e., maxima or minima) need to be predicted. This is the case for minimal clock cycle prediction, but also for the prediction of hot spots during routing (maximal routing congestion).

Recently, some research results have been published that indicate that the size of a net's neighborhood is an important factor in its degree of optimization [BN01, YM03]. Several neighborhood metrics can be used. In general, they indicate how many gates are topologically close to the net and are therefore preferably placed close to each other. The idea is that the more gates that need to be placed close together, the more difficult the job of the placement tool will be and the worse its performance. Figure 7.1 shows some results that indicate that, when the total source-to-sink length is optimized during placement, this hypothesis is only valid to a certain extent. We have measured the average wire length across 100 `Plato` placements for all source-to-sink connections in benchmarks `dsip` and `rot`. For each of those connections, we have also counted the total number of other source-to-sink connections attached to its source and sink (the size of the environment). In Figure 7.1, the relationship between both is shown. It seems that the average placement length of a connection only increases drastically for very large net environments.

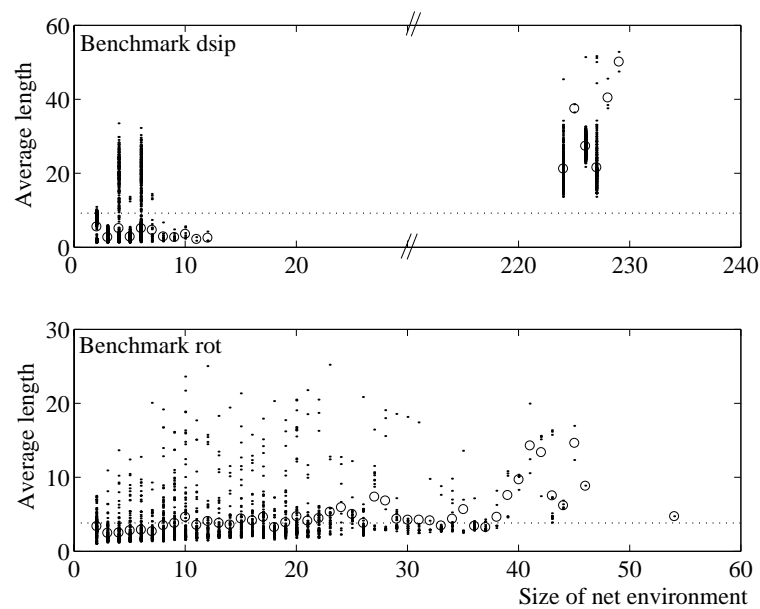


Figure 7.1: Average lengths across 100 `Plato` placements of all source-to-sink connections in benchmarks `dsip` and `rot` as a function of the size of their environment (dots). The circles indicate the average across all nets with the same environment size, while the dotted line shows the average length across all wires.

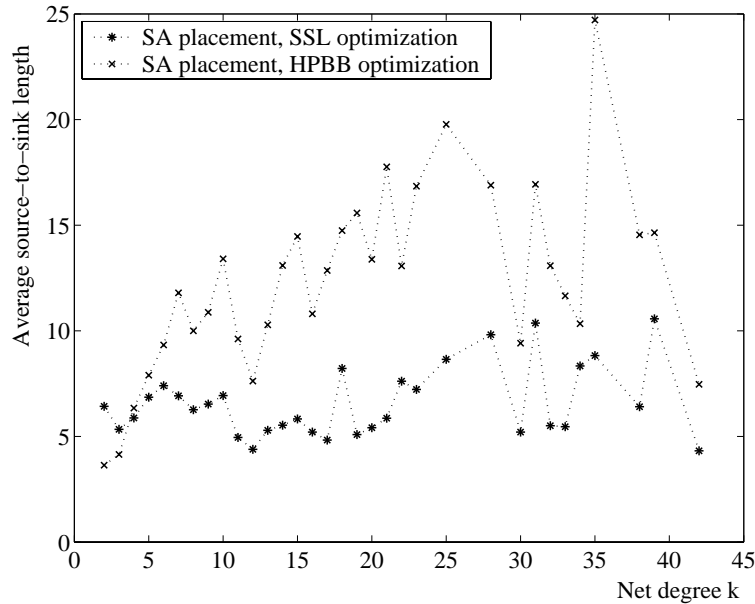


Figure 7.2: Average source-to-sink lengths of SA placements of benchmark *ibm01*, resulting from a placement in which total SSL was optimized and one where total HPBB-length was optimized.

Applied to source-to-sink connections, a similar hypothesis could be that the larger the degree of the net a source-to-sink connection originates from, the longer it will turn out after placement (on average). In Figure 7.2, the average length of the source-to-sink connections in benchmark *ibm01* are plotted as a function of the degree of the net they originate from. These data were obtained from two SA placements of the original benchmark (i.e., *with* multi-terminal nets). In the first placement, the total source-to-sink length (SSL) was optimized. The source-to-sink lengths resulting from this placement are the same as when the transformed benchmark (with two-terminal nets only) is placed. The resulting average lengths show the same trends as for benchmarks *dsip* and *rot*: the average length of source-to-sink connections does not increase significantly, except for very large-fanout nets.

However, in the second placement, the cost function was the total half-perimeter bounding box (HPBB) length. For this placement, the average length of source-to-sink connections increases much more rapidly. This is due to the nature of HPBB optimization: as long as the bounding box does not change, the position of the terminals within the

bounding box does not matter. Hence, because they are not *visible* in the optimization cost function, they are not further optimized and can be assumed to be random. As a result, the homogeneity assumption is no longer fulfilled. As the net degree increases, relatively fewer source-to-sink pairs are visible in the cost function, so the average source-to-sink length for large-fanout nets is larger than that for small-fanout nets.

This is also illustrated in Appendix C, where we derive a strategy to approximate the length distribution of multi-terminal nets from the length distribution of source-to-sink connections *under the homogeneity assumption*. These approximations are relatively good when SSL is optimized during placement, but totally wrong when HPBB-length is optimized (Figure 7.3), because our approximation technique applies the global source-to-sink length distribution to all net degrees.

We can conclude that, for circuits with multi-terminal nets, the cost function that is optimized during placement has a very large impact on (the homogeneity of) the resulting source-to-sink lengths. As a consequence, its impact must be incorporated in future prediction flows if we are to achieve some level of accuracy.

Thus far, the most solid work on the inclusion of multi-terminal nets in recursive partitioning-based interconnect prediction models was done by Stroobandt [Str01a, Str01b]. However, his work is still situated within Donath's original framework, using Rent's rule and balanced four-way partitioning. Quite probably, elements from Stroobandt's approach, our extended hierarchical prediction model and the derivation of HPBB-site functions given in Appendix C can be combined to obtain highly accurate predictions of multi-terminal wire length distributions.

7.1.3 Other extensions to be considered

One of the topics that urgently needs to be studied is the impact of routing and routing congestion on wire lengths. It is well known that due to routing congestion, small routing detours must sometimes be made. Also, to avoid such congestion, the gates in candidate congested areas are sometimes placed further apart than would be possible if total placement wire length were the only optimization objective. Obviously, compared to routed wire lengths, our predictions will be less reliable than compared to placement wire lengths. However, to our knowledge, the extent of this loss of accuracy has not yet been thoroughly studied. Such a study is also necessary to determine how and to what extent the

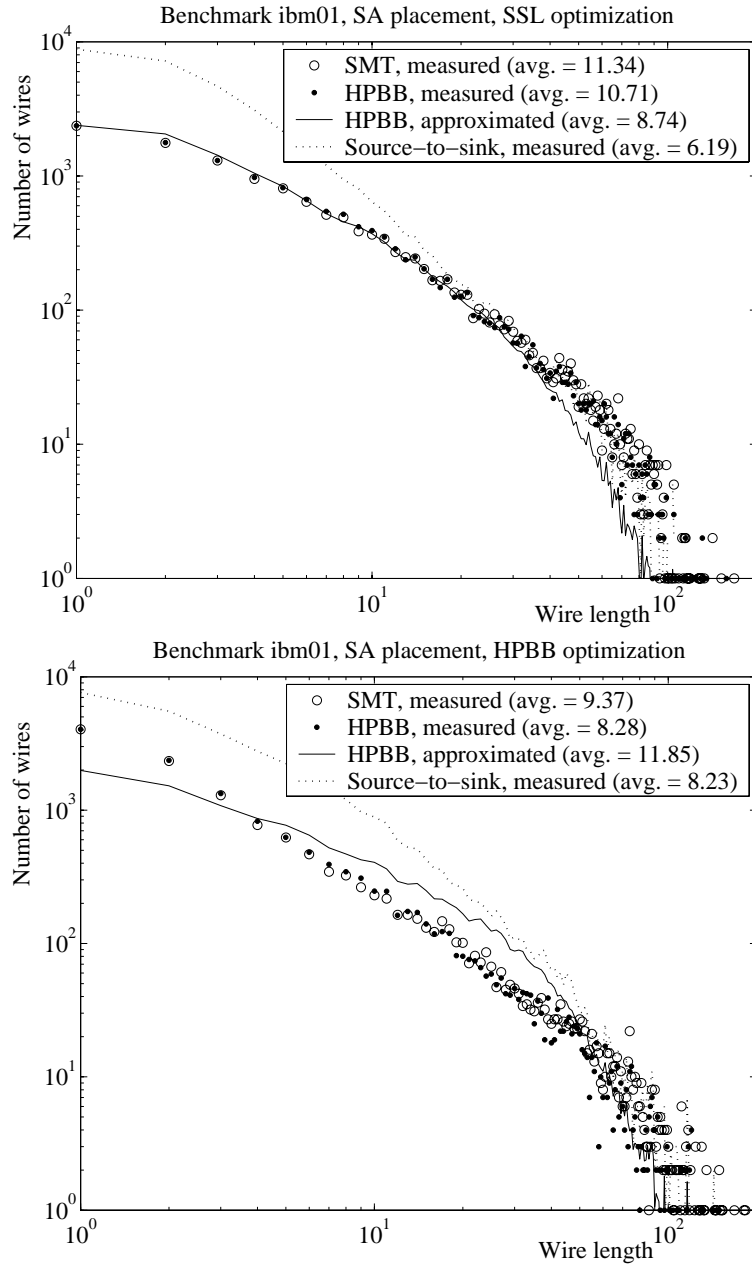


Figure 7.3: Results from two SA placements of the multi-terminal net version of benchmark *ibm01*: (top) measured Steiner minimal tree (SMT) length, half-perimeter bounding box (HPBB) length and source-to-sink length (SSL) distributions and approximated HPBB length distribution when SSL is optimized during placement; (bottom) measured SMT, HPBB and source-to-sink length distributions and approximated HPBB length distribution when HPBB-length is optimized during placement.

current prediction models can be extended to include routing effects.

Another topic for further study is the impact of different types of heterogeneity in circuits. A first type of heterogeneity regards the interconnection complexity. Some circuits consist of parts with different interconnection complexity. The prediction of wire lengths for such circuits has only sporadically been addressed in literature (e.g., [VMSVC95b, ZH01]). In his Ph.D. thesis [Ver03], Verplaetse studied this topic somewhat more thoroughly. He described different types of heterogeneity in the interconnection complexity and studied whether the different parts could still be extracted from a flattened netlist. He also derived a stochastic model for the terminal-gate relationship that includes that variance on the number of terminals and the way that variance is propagated through successive levels of recursive partitioning. However, the impact of such heterogeneity on placement quality and the best way to model it in a wire length prediction model still require further investigation.

A second type of heterogeneity is that of strongly differing block sizes. In ever more circuit designs, large IP (*intellectual property*) blocks are used. These are often pre-designed and much larger than regular cells. Also, there is a trend to use an increasing number of relatively small embedded memory blocks instead of one large memory. These blocks are also usually larger and have different shapes than normal library cells. Again, the question arises whether these need to be incorporated in prediction models and how this is best done. In [CKLS02], site functions for placement grids with forbidden regions (caused by, e.g., pre-placed IP-blocks) are calculated and used for prediction in Davis' technique. To our knowledge, no other results on this topic have been published so far.

Finally, with respect to wire length distributions, we can also identify a third kind of heterogeneity, caused by the different optimization efforts that are spent on each net. These can be due to, for instance, different weights assigned to the nets to achieve clock cycle optimization.

The minimal clock cycle is only one of the design cost metrics that depend on interconnect length. Other such metrics are, e.g., yield and power dissipation. For most derived metrics, the predicted wire length distribution needs to be combined with other properties of the circuit (behaviour or topology) or the design strategy.

For instance, the power consumption caused by a wire depends on its length, but also on how frequently signal transitions occur on

that wire (its *toggle rate*). In a power-optimized design, the wires with the highest toggle rates are the most critical [AN03]. Again, the question arises regarding the relation between optimization strength and wire length distribution, given the criticality distribution. Furthermore, since toggle rates are properties of individual wires, predictions of the power dissipation can be expected to be sensitive to local deviations from the homogeneity assumption (as was the case with clock cycle prediction).

In functional yield prediction, the goal is to predict the fraction of chips that will be faulty. With respect to interconnect, faults can occur in the form of *bridges* (two neighbouring wires that are accidentally connected) or *cuts* (broken wires). The probability that such faults occur depends (amongst other things) on the wire length distribution [CP01], but also to a large extent on the routing strategy, which is not yet included in our models.

7.2 Downwards: deeper into the theoretical foundations

This Ph.D. thesis has resulted in an increased insight in interconnect prediction models, which has resulted in better predictions of placement wire length distributions. However, we still need to rely on recursive circuit partitioning to extract the relevant topological parameters to be used in our models. This way of characterizing circuits is in fact indirect, in that we need to perform one type of combinatorial optimization (circuit partitioning) to be able to predict the results of another, i.e., circuit placement. The reason why we need to perform this indirect measurement, is that we still do not know what exactly in the circuit topology causes the optimization results to be what they are.

Obviously, it would be far more desirable to be able to do some quick and direct measurements on the circuit topology. Also, if interconnect properties are to be optimized during the design stages where the graph topology is actually created, i.e. R/T and logic level synthesis and optimization, we need to know how to optimize them, so we need to get more insight into the link between interconnect patterns and the achievable degree of optimization. A very important question in this context is whether it is at all possible to characterize the circuit as a whole by performing some fast and local measurements. And if

the answer is positive, how is this to be done?

In this section, we first illustrate the close relationship between placement and partitioning optimization in terms of the solution space that needs to be explored for both problems. In particular, this analysis links the respective solution spaces to local and ever more global interconnection patterns in the circuit graph. Then, we briefly explore other research domains that involve graphs, in search of possible starting points for answering our own questions. We first discuss random graph theory, which is by far the most mature area of graph research. Then, we switch to graphs that occur as a representation of various types of relationships and the way they are characterized. Finally, we review the few existing theoretical results on optimal graph bipartitioning.

7.2.1 Why a partitioning based placement model works

Donath's technique for interconnect prediction mimics a real placement procedure and uses measurable properties of the circuit topology. Since the use of the Rent characteristic in that model results in an even closer match between what a partitioning-based placement tool actually does and our predictions, it is no surprise that the results are better.

However, this does not explain why the circuit partitioning properties can be used to predict, e.g., the level of optimization that can be reached by swapping (cfr. Verplaetse's occupation probability in Chapter 5), or why the average wire lengths from partitioning-based placement tools correlate so well with the results from a flat placement tool such as SA. Previous explanations (e.g., [Str01b]), focus on the self similarity that is intrinsic to a power law (i.e., Rent's rule). However, they do not offer an explanation for the reason why the results of one optimization (partitioning) are so closely related to the results of another (placement). The circuit partitioning Rent characteristic must indirectly measure some essential properties that characterize the behavior of a circuit's interconnect topology under placement optimization. In what follows we will give some indications of why we believe that this is indeed the case.

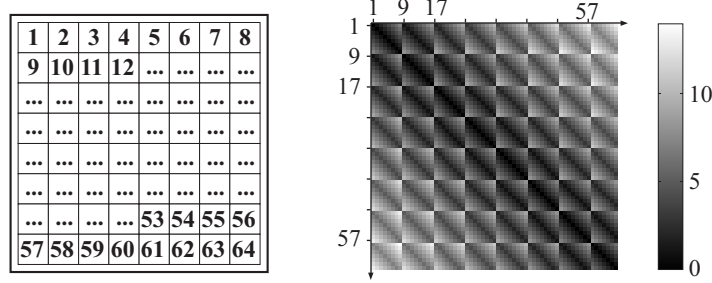


Figure 7.4: Graphical representation of the numbering of the grid positions and the corresponding matrix of costs c_{ij} for a 8x8 placement grid with square cells.

A different view on placement optimization

In this section, we look at the placement optimization of circuits with two-terminal nets only in a slightly different way to illustrate the impact of the placement architecture and the circuit topology on the placement cost function. Consider a placement process that generates n random placements and returns the best of these n (i.e., one of those with the smallest total wire length C). Obviously, this is not a feasible strategy, since to obtain a really good result, huge numbers of placements would need to be generated. However, it can be used to get more insight into the search space for placement optimization.

For randomly generated placements, it is possible to calculate the moments of the pmf of C . Whereas, for higher order moments, this would require a huge amount of computation time, it is relatively easy for the first and second moments.

Consider assigning a unique number to each grid location (e.g., in a 2-D grid, incremental from left to right and from top to bottom as in Figure 7.4). Now let c_{ij} ($i, j = 1 \cdots X_g Y_g$) be the Manhattan distance between grid positions i and j and w_{ij} the number of (two-terminal) nets between grid positions i and j in a given placement. With these conventions, we can rewrite C as:

$$C = \sum_{i=1}^{X_g Y_g} \sum_{j=i+1}^{X_g Y_g} c_{ij} w_{ij}. \quad (7.1)$$

To simplify the notation, we use \sum_{ij} as a shorthand for

$$\sum_{i=1}^{X_g Y_g} \sum_{j=i+1}^{X_g Y_g} . \quad (7.2)$$

Note that, for placement experiments where the cost function is the weighted sum of all wire lengths, Equation (7.1) is still valid, but the w_{ij} become the sum of the net weights of all nets connecting grid locations i and j .

If we now perform random placements of a given circuit with a total of N_c internal nets, then the w_{ij} become random variables with

$$\mu_w = E[w_{ij}] \simeq \frac{\sum_{ij} w_{ij}}{S_g} = \frac{N_c}{S_g}, \quad (7.3)$$

$$\sigma_w^2 = E[w_{ij}^2] - \mu_w^2 \simeq \frac{\sum_{ij} w_{ij}^2}{S_g} - \mu_w^2. \quad (7.4)$$

where S_g is the total number of two-terminal net placement sites in the grid.

Since the probability of selecting any specific pair of grid locations is equal to $1/S_g$, we find for the site distribution $\mathcal{D}(\ell)$:

$$\mu_{\mathcal{D}} = \frac{\sum_{ij} c_{ij}}{S_g}, \quad (7.5)$$

$$\sigma_{\mathcal{D}}^2 = \frac{\sum_{ij} c_{ij}^2}{S_g} - \mu_{\mathcal{D}}^2. \quad (7.6)$$

Using Equations (7.3)-(7.4) and (7.5)-(7.6), the mean and variance of the cost C for random placements of a circuit can be calculated:

$$\begin{aligned} \mu_C &= E\left[\sum_{ij} c_{ij} w_{ij}\right] \\ &= \sum_{ij} c_{ij} E[w_{ij}] \\ &= S_g \mu_{\mathcal{D}} \mu_w \\ &= N_c \mu_{\mathcal{D}} \end{aligned} \quad (7.7)$$

$$\sigma_C^2 = E\left[\left(\sum_{ij} c_{ij} w_{ij}\right)^2\right] - \mu_C^2$$

$$\begin{aligned}
&= \sum_{ij} c_{ij}^2 E[w_{ij}^2] + \sum_{ij} \sum_{pq \neq ij} (c_{ij} c_{pq} E[w_{ij} w_{pq}]) - \left(\sum_{ij} c_{ij} \right)^2 \mu_w^2 \\
&= \sum_{ij} c_{ij}^2 E[w_{ij}^2] + \sum_{ij} \sum_{pq \neq ij} (c_{ij} c_{pq} E[w_{ij} w_{pq}]) \\
&\quad - \left(\sum_{ij} c_{ij}^2 + \sum_{ij} \sum_{pq \neq ij} c_{ij} c_{pq} \right) \mu_w^2 \\
&= \sigma_w^2 \sum_{ij} c_{ij}^2 + \sum_{ij} \sum_{pq \neq ij} (c_{ij} c_{pq} \cdot Cov[w_{ij} w_{pq}]) \\
&= S_g \sigma_w^2 (\sigma_D^2 + \mu_D^2) + \sum_{ij} \sum_{pq \neq ij} (c_{ij} c_{pq} \cdot Cov[w_{ij} w_{pq}]) \quad (7.8)
\end{aligned}$$

There are two distinct cases for $Cov[w_{ij} w_{pq}]$: the net sites indicated by indices ij and pq can either be adjoining or not adjoining. In the first case, both net sites have one grid location in common, i.e., $i = p, j \neq q$. However, because we are considering random placements, we can assume $Cov[w_{ij} w_{iq}]$ to be independent of the specific values of i, j and q , respectively. Hence, we can substitute $Cov[w_{ij} w_{iq}]$ by a constant value $s_{w,a}^2$, the covariance for adjoining net sites. In the second case, the net sites are between four distinct grid locations: $i \neq j \neq p \neq q$. Again, we can assume the covariance for these sites to be independent of the site locations. We will refer to the covariance for non-adjoining net sites as $s_{w,n}^2$. Since we are considering random placements and nets with unit weight, both covariances can be measured from the graph topology by counting the number of nets (for $s_{w,n}^2$) and the number of distinct net pairs with one block in common (for $s_{w,a}^2$).

Equation (7.8) can now be rewritten as:

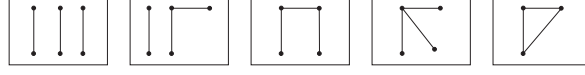
$$\sigma_C^2 = S_g \sigma_w^2 (\sigma_D^2 + \mu_D^2) + \left(s_{w,a}^2 \sum_{i \neq j \neq q} c_{ij} c_{iq} + s_{w,n}^2 \sum_{i \neq j \neq p \neq q} c_{ij} c_{pq} \right) \quad (7.9)$$

In this equation, the summations across $c_{ij} c_{iq}$ and $c_{ij} c_{pq}$, respectively, can be considered as sums across *partial second order site functions*. They are partial in the sense that they each represent only a fraction of all net site pairs in the grid.

Because the average wire length is given by C divided by the number of nets N_c , it follows that

$$\mu_\ell = \frac{\mu_C}{N_c} = \mu_D \quad (7.10)$$

Interconnection patterns with 3 nets:



Interconnection patterns with 4 nets:

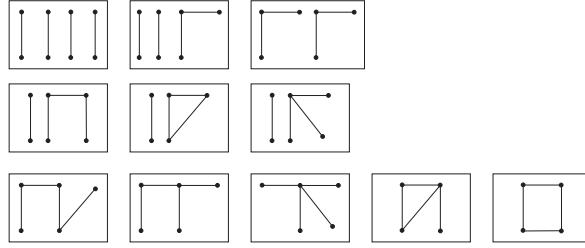


Figure 7.5: Different interconnection patterns with three and four nets.

$$\sigma_\ell^2 = \frac{S_g \sigma_w^2 (\sigma_D^2 + \mu_D^2)}{N_c^2} + \frac{s_{w,a}^2}{N_c^2} \sum_{i \neq j \neq q} c_{ij} c_{iq} + \frac{s_{w,n}^2}{N_c^2} \sum_{i \neq j \neq p \neq q} c_{ij} c_{pq} \quad (7.11)$$

A similar strategy can be followed to calculate the higher order moments and central moments of the pmf for C . Whereas for the second moment (and the standard deviation) we only need second order covariances, higher order covariances are required for the higher order moments. They too can be determined from the graph topology, but the number of different interconnection patterns that need to be counted rapidly increases for higher orders. To illustrate this, Figure 7.5 shows all different interconnection patterns that need to be considered to compute the third and fourth moments.

Hence, as might be expected, the pmf of C for random placements is entirely determined by the circuit topology and the placement grid. As a result, the placement with minimal cost that is found by a particular placement tool is also determined by properties of the interconnect topology, in particular the occurrence of different interconnect patterns in the graph.

Now consider the class of uniform random circuits, generated by creating N_c nets, each connecting a node pair that was randomly selected with uniform probability. On average, across many individual circuits from this class, there is no correlation between the number of nets between any two node pairs (and hence grid locations). The co-

variances $s_{w,a}^2$ and $s_{w,n}^2$ are equal to zero, and we find:

$$\sigma_\ell^2 = \frac{S_g \sigma_w^2 (\sigma_D^2 + \mu_D^2)}{N_c^2}. \quad (7.12)$$

This expression only depends on the circuit size, its number of nets and the placement grid.

Relationship between placement and partitioning

The derivations in the previous section, performed for placement, can easily be extended to partitioning by generalizing the c_{ij} to signify the cost of a single connection between locations i and j . For standard k -way partitioning, this cost is 0 if i and j are in the same module, and 1 if they are in different modules.

We will replace S_g , the number of grid locations, by S_p , the number of partitioning *slots*. In strictly balanced partitionings, S_p equals G_c , the number of gates in the circuit, and the number of slots in each module equals G_c/k . The larger S_p , the more imbalance is allowed in the partitioning. Similar to the site function, we can define the module function \mathcal{M} as the cost distribution for randomly partitioned modules. Its mean and variance are now given by:

$$\mu_{\mathcal{M}} = \frac{\sum_{ij} c_{ij}}{S_p}, \quad (7.13)$$

$$\sigma_{\mathcal{M}}^2 = \frac{\sum_{ij} c_{ij}^2}{S_p} - \mu_{\mathcal{M}}^2. \quad (7.14)$$

The total partitioning cost is given by:

$$\mu_C = N_c \mu_{\mathcal{M}}, \quad (7.15)$$

and its variance by:

$$\sigma_C^2 = S_p \sigma_w^2 (\sigma_{\mathcal{M}}^2 + \mu_{\mathcal{M}}^2) + s_{w,a}^2 \sum_{i \neq j \neq q} c_{ij} c_{iq} + s_{w,n}^2 \sum_{i \neq j \neq p \neq q} c_{ij} c_{pq}. \quad (7.16)$$

Now, instead of the average wire length, we can derive the mean and variance of the average number of terminals T per module, which is given by

$$T = \frac{2}{k} C. \quad (7.17)$$

We can also derive the fraction of nets that is cut, α_{rand} , which is given by:

$$\alpha_{rand} = \frac{C}{N_c}. \quad (7.18)$$

We can conclude that indeed, the search space for partitioning is determined by the same topological properties, the frequencies of occurrence of certain interconnect patterns, as the search space for placement optimization. Therefore, it would be very useful to direct further research at the modeling of these frequencies and their relationship to the optimal (placement or partitioning) result that is found by different tools.

Extensive research on this topic has already been done for uniform random graphs. We briefly summarize the main results from this research domain in the next section.

7.2.2 Random graph theory

The foundations of Random graph theory were laid by Erdős and Rényi [Erd59, ER60]. They described the statistical properties of random graphs $\mathcal{G}(n, p_E)$, with n nodes and an edge probability $p_E(n)$, usually expressed as a function of n .¹ The random graph $\mathcal{G}(n, p_E)$ must not be interpreted as a specific individual graph, but rather as a probabilistic result of a randomized graph generator, which generates an edge between each node pair with *uniform* probability p_E . Note that many other randomized graph generators can be defined. Indeed, several such generators are used to mimic the real-world graphs described in the next section. However, the term *random graphs* is generally reserved for those graphs generated with a uniform edge probability.

In total, a random graph $\mathcal{G}(n, p_E)$ has $2^{\binom{n}{2}}$ possible graph instances $G(n, e)$ on n nodes. Every individual graph instance $G(n, e)$ with e edges has an equal probability

$$(p_E)^e (1 - p_E)^{[n(n-1)/2] - e}$$

of being generated by that graph generator. Hence, the total number of edges in $\mathcal{G}(n, p_E)$ is a random variable that follows a binomial distribu-

¹The edge probability p_E is not to be confused with the Rent exponent p . Both symbols are well established in literature and it would seem very artificial to introduce new symbols here to avoid conflict. Hence, we have only added the subscript in p_E to make the distinction.

tion with expectation

$$E[e] = p_E n \frac{n-1}{2}. \quad (7.19)$$

Random graph theory addresses the joint average properties of this set of graph instances and the scaling of these properties as a function of n and $p_E(n) \sim n^{-z}$. Graphs for which $z = 0$ (p_E independent of n) are called *dense random graphs*. For these graphs, the average number of edges connected to a node (the average *node degree*) increases linearly with n . Another class of random graphs that is often studied is that of *sparse random graphs*, for which $p_E \sim 1/n$ and the average node degree is independent of n .

A random graph $\mathcal{G}(n, p_E)$ is said to have a certain property or *attribute* A , if *almost every graph instance* in $\mathcal{G}(n, p_E)$ has this property as n grows very large. Mathematically, this is expressed as:

$$\lim_{n \rightarrow \infty} P(A) = 1, \quad (7.20)$$

It turns out that, for many random graph attributes, there exists a threshold value z for the scaling behavior for $p_E(n) \sim n^{-z}$. If $p_E(n)$ is smaller than that threshold (i.e., if it decreases faster than the threshold scaling function), the limit in Equation (7.20) equals zero and if it is larger, this limit equals one. The existence of such a threshold is also referred to as a *zero-one law*. One class of attributes for which zero-one laws exist is the presence of specific interconnect patterns on a given number of nodes, such as the ones shown in Figure 7.6. For instance, edge pairs with a common vertex (starting at $p_E \sim n^{-3/2}$), triangles and all cycles (starting at $p_E \sim n^{-1}$) or fully connected subgraphs of four vertices (starting at $p_E \sim n^{-2/3}$). In terms of statistical physicists, the presence of such a threshold is also referred to as a *phase transition*.

Now consider a particular interconnect pattern F of k nodes and l edges. In random graphs $\mathcal{G}(n, p_E)$, the expected number of occurrences X of this interconnect pattern [Bol85] is:

$$E[X] = C_n^k \frac{k!}{a} p_E^l, \quad (7.21)$$

where a is a correction factor to account for the number of interconnection patterns that are isomorphic to F .

We can conclude that, for random graphs, the average interconnect topology, characterized by the occurrence of certain interconnection patterns is completely determined by the graph size n and the edge probability p_E .

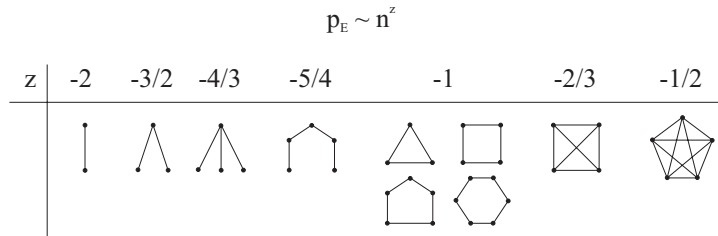


Figure 7.6: Threshold scaling laws for the appearance of some typical interconnect patterns in random graphs.

7.2.3 Topological properties of non-random graphs

Besides for representing VLSI circuits, graphs are used as a representation for many different kinds of systems. In the research domains where these graphs are used, various models for graph characterization have emerged. Also, in many of those fields, including that of VLSI design [Str01b, Ver03], those models have been applied to synthesize new, typical graphs or *benchmarks*, that can be used for the study and evaluation of, e.g., domain specific optimization algorithms. It is reasonable to assume that an exchange of experience and ideas between these different worlds would be fruitful.

Recently, a strongly interdisciplinary research domain has emerged, combining results and techniques from, e.g., traditional random graph theory and statistical mechanics with properties and observations from several application domains. Examples of networks studied in this context are the world wide web (i.e., the network of linked web-pages) and the Internet (the underlying physical network), social networks, neural networks and the system of interacting agents in biological metabolic systems. Some conditions that are examined are the impact of random node defects on the Internet or the spreading of sexually transmitted diseases in (sexual) social networks.

However, thus far, the existing results and models concerning circuit graphs have not been included in this interactive domain. This is probably due to the fact that only very few researchers from the VLSI research community have found the way to those other domains of graph research. In the field of circuit graphs, most of the characterization work has focused on the partitioning properties [Str01b, Ver03] or on models for the degree distribution of multi-terminal nets [Str01b, ZHDLM00]. In his Ph.D. thesis [Ver03], Verplaetse compared the parti-

tioning properties of VLSI circuit graphs to those of several other graph types and in [CKL01], an attempt was made to link the growth models for network graphs from [AB02] to Rent's rule and the existing models for the degree distribution of multi-terminal nets.

In the interdisciplinary network community, it was soon discovered that the topological properties of graphs representing many real world systems significantly deviate from those of random graphs. For many years large efforts have been invested in trying to capture the topological properties of the specific graphs or *networks* encountered in specific research domains. In most cases, the aim was to try and predict the global properties of the system, as well as its behavior under certain conditions, based on statistical models for the local interactions (*edges*) between network units (*nodes*). Therefore, we believe that a more extensive study of the models applied in other graph research domains might lead to additional insight and better graph characterization models.

For many graphs corresponding to the interactions in real world systems, several topological properties were identified to differ significantly from those of random graphs. We will now study the most important graph properties that are defined in the context of complex networks and compare results from the literature to those for circuit graphs. For complex networks, three important types of topological properties are extracted (descriptions based on [AB02]):

The *clustering coefficient* expresses the presence of groups of densely interconnected nodes, disregarding edge direction. In this context, a *cluster* (or *clique*) is defined as a set of nodes that are fully interconnected. If a node i with node degree k_i , i.e. connected to k_i other nodes,² is part of a cluster, there are $k_i(k_i - 1)/2$ edges between its k_i nearest neighbors. The clustering coefficient C_i of node i is the ratio between the actual number of edges E_i between these k_i nodes and the value $k_i(k_i - 1)/2$, i.e.,

$$C_i = \frac{2E_i}{k_i(k_i - 1)}. \quad (7.22)$$

²Multiple connections between the same node pair are not considered in this formalism: an edge exists between two nodes if there is a certain relationship between them. Multiplicity, if at all present, is expressed by assigning weights to the edges. Also, in contrast to circuit graphs, external edges are not considered. Hence, they are also not present in the node degree.

The clustering coefficient C for the graph is the average of all individual C_i 's. In a random graph, the graph clustering coefficient approximately equals p_E , i.e. the probability that an edge is present between a randomly selected node pair. In most types of real networks that have been studied so far, C is much larger than in a random graph with the same number of nodes and edges, indicating a certain systematic *locality* in the interconnect pattern.

The *path length*, or graph distance between two nodes is defined as the minimal number of edges that have to be traversed to connect them. Here, the distinction is often made between directed and undirected graph representations of the same network. Indeed, for the same graph topology, disregarding edge direction usually results in much shorter path lengths. Both average and maximal path length are reported in literature as relevant parameters. For the average path length ℓ , the average is taken across all node pairs. The maximal path length for a node is the maximal graph distance between that node and any other node in the graph. Hence, both the *average maximal path length* across all nodes, as well as the global maximal path length or the graph *diameter* can be considered.³ For random, undirected graphs, the graph diameter was shown to be:

$$d_{rand} = \frac{\ln(G_c)}{\ln(\langle k \rangle)} \quad (7.23)$$

with $\langle k \rangle$ denoting the average node degree ($=t'_G$). It is assumed that the average path length ℓ_{rand} also scales accordingly, i.e., $\ell_{rand} \sim d_{rand}$.

Graphs that have a high clustering coefficient as well as short path lengths are often called *small worlds*.

The *node degree distribution* is the histogram of all node degrees in the graph. In directed graphs, the distinction is often made between a node's *indegree* (i.e., the number of incoming edges) and its *outdegree* (the number of outgoing edges). Note that, since hypergraphs are rarely used in this context, all fanout is due to the nodes' outdegrees. The node degrees in undirected random

³Again, the terminology in this section was mostly copied from [AB02]. As such, it is not always consistent with the terminology used in this thesis. However, since it is confined to this chapter, we hope to avoid confusion as much as possible.

Network	Size	$\langle k \rangle$	γ	ℓ	d_{rand}	C	C_{rand}
(a) WWW (sites - 1999)	153,127	35.21		3.1	3.35	0.1078	2.3×10^{-4}
(b) WWW (pages - 2000)	2×10^8	7.5	2.71/2.1	16	8.85		
(c) Movie actors (1998)	225,226	61		3.65	2.99	0.79	2.7×10^{-4}
Movie actors (1999)	212,250	28.78	2.3	4.54	3.65		
(d) Neuroscientists	209,293	11.5	2.1	6	5.01	0.76	5.5×10^{-5}
(e) English words	460,902	70.13	2.7	2.67	3.03	0.437	10^{-4}

Table 7.1: Topological parameters for some very different networks (taken from [AB02]). For network (b), both γ_{out} and γ_{in} (in that order) were given. Note that the two editions of network (c) were reported by different authors who may have extracted their networks in a slightly different way. This might explain why the network size appears to have decreased from 1998 to 1999.

graphs follow a binomial distribution. For large graphs and a sufficiently high number of edges, this approaches a Poisson distribution:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!} \quad (7.24)$$

In many real graphs however, this distribution is much closer to a power law, characterized by its exponent γ , i.e.:

$$P(k) \sim k^{-\gamma} \quad (7.25)$$

Often, it only has a power-law tail (i.e. only for sufficiently large node degrees) that is cut off at a given node degree value. For directed graphs, these power-law tails have been reported both for the indegree distribution, $P(k_{in}) \sim k_{in}^{-\gamma_{in}}$ and the outdegree distribution $P(k_{out}) \sim k_{out}^{-\gamma_{out}}$. Networks with a power-law degree distribution are said to be self-scaling, referring to the self-scaling properties of power law functions.

The values of these three parameters have been measured and reported in literature for a broad range of different networks. A few of these results are given in Table 7.1. Note that all data reported in this table were taken from the survey paper [AB02], and not from their original sources.

Networks (a) and (b) were derived from the world wide web in 1999 and 2000, respectively. In (a), nodes represent entire web sites, while in (b), they correspond to individual web pages. In both cases,

a directed edge is present between two nodes if there exists a hyper-link between the corresponding web sites or web pages, respectively. Network (c) represents the movie actor collaboration network, based on the Internet Movie Database (all movies and their casts since the 1980s) at two points in time (1998 and 1999). Here, nodes correspond to actors and an (undirected) edge between two nodes indicates that they have acted in a movie together. Similar networks have been derived for collaborations between scientists in different specialization areas, with nodes now representing scientists and edges indicating that they have co-authored at least one article. Network (d) is an example for neuroscientist publications between 1991 and 1998. Finally, network (e) was constructed from the British National Corpus with English words as nodes. They are connected by an edge if they appear next to each other or separated by one other word in sentences.

The general conclusion from these data is that many networks that originate from real-world systems show small world behavior as well as power-law degree distributions. Their average path lengths are usually of the same order of magnitude as the predicted diameters of random graphs of the same size and average degree. Their clustering coefficients, however, are several orders of magnitude above those of their random counterparts.

We were curious to see whether this is also the case for circuit graphs. Hence, we have measured the topological graph properties for all of the ISPD98 benchmarks. For this purpose, we used the two-terminal versions of the circuits, and disregarded all external connections. The results are reported in Table 7.2. Figure 7.7 shows a typical example of a node degree distribution (benchmark circuit `ibm15`, $\gamma = 3.074$). It has long been known that the net degree distribution of circuit graphs can also be approximated by a power-law [Str01b]. Since in our circuit graphs, all net fanout has been pushed back into the nodes, it is not very surprising to find a similar distribution for the node degrees.

Overall, we find that the topological properties of the ISPD98 circuit graphs and the ways in which they deviate from random graph properties are very similar to what has been reported for other graphs. This general phenomenon is quite surprising, given the very different ways in which all these different types of graphs were generated.

benchmark	Size	$\langle k \rangle$	γ	ℓ	d	d_{rand}	C	C_{rand}
ibm01	12,506	5.61	4.52	8.60	19	5.47	0.108	4.49e-4
ibm02	19,342	5.66	N.A.	6.65	14	5.69	0.045	2.93e-4
ibm03	22,853	5.43	2.92	7.10	14	5.93	0.100	2.38e-4
ibm04	27,219	5.11	2.70		24	6.26	0.052	1.88e-4
ibm05	28,146	6.57	N.A.	7.37	17	5.44	0.028	2.33e-4
ibm06	32,332	5.64	3.38	7.79	16	6.00	0.031	1.75e-4
ibm07	45,639	5.47	4.65	9.48	20	6.31	0.068	1.20e-4
ibm08	51,022	5.52	4.52		40	6.35	0.074	1.08e-4
ibm09	53,110	5.78	3.02	9.37	21	6.20	0.107	1.09e-4
ibm10	68,685	6.16	5.35	9.71	24	6.13	0.061	8.97e-5
ibm11	70,152	5.39	3.34	9.87	21	6.62	0.051	7.68e-5
ibm12	70,439	6.66	5.16		21	5.89	0.047	9.45e-5
ibm13	83,709	5.80	3.15		19	6.45	0.061	6.92e-5
ibm14	147,088	5.26	4.64	10.90	24	7.16	0.042	3.58e-5
ibm15	161,185	6.31	3.07	9.44	29	6.51	0.056	3.91e-5
ibm16	182,980	6.25	5.13	11.18	21	6.61	0.045	3.42e-5
ibm17	184,752	7.11	5.15	10.05	22	6.18	0.038	3.85e-5
ibm18	210,341	5.52	4.13	9.75	25	7.17	0.036	2.62e-5

Table 7.2: Topological parameters for circuit graphs corresponding to the two-terminal net versions of the ISPD98 benchmark circuits. For benchmark circuits `ibm02` and `ibm05`, the node degree distribution does not follow a power law. Benchmark `ibm02` has a bimodal node degree distribution, showing a second maximum at far higher node degrees. For benchmark `ibm05`, the highest occurring node degree is relatively small, leaving insufficient data points to conclude that a power-law tail is present.

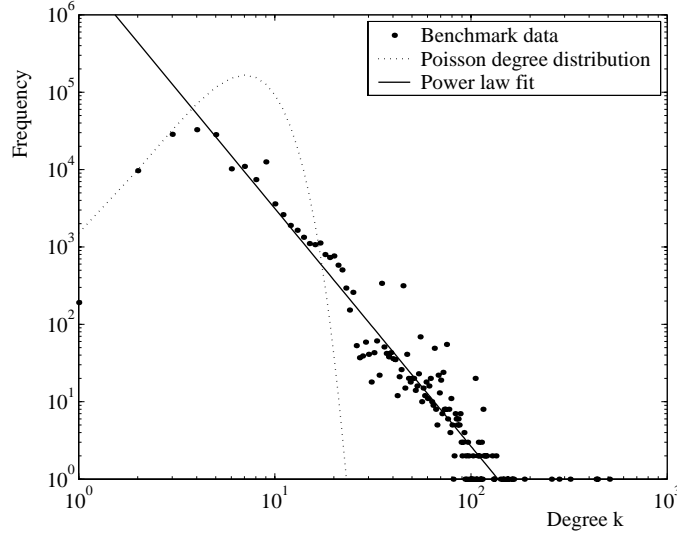


Figure 7.7: Node degree distribution for circuit graph corresponding to benchmark circuit `i_bml15`. Also shown are the corresponding random graph (Poisson) node degree distribution and the power-law fit resulting in $\gamma = 3.074$.

7.2.4 Optimal bipartitioning of random and non-random graphs

In the 1980's, researchers from the statistical mechanics community realized that the statistical mechanics of random systems might well be applicable to predict the result of combinatorial optimization problems on random graphs. In [FA86], Fu and Anderson studied the application of statistical mechanics to NP-complete problems in combinatorial optimization. In particular, they treated the statistical mechanics of graph bipartitioning for two classes of random graphs. For the first class, that of dense random graphs, they found the mean minimal cut size $E[C_l]$ to be:

$$E[C_l] = \frac{p_E n^2}{4} - 0.3816 \sqrt{p_E(1-p_E)} n^{3/2} + O(n). \quad (7.26)$$

With the total number of edges given by Equation (7.19), the corresponding cut probability scaling law is given by:

$$E[\alpha_0] = \frac{1}{2} \frac{n}{n-1} - 0.7632 \frac{\sqrt{n}}{n-1} \sqrt{\frac{1-p_E}{p_E}} + O(1/n) \quad (7.27)$$

Note that the techniques they applied to obtain this result are valid only in the *thermodynamic limit*. Though they admit that the meaning of this

concept is not clear in an optimization problem, it does imply that the system must be sufficiently large. As n goes to ∞ , only the first term in Equation (7.27) remains, i.e., for very large dense random graphs, an optimized partitioning is no better than a random partitioning. For large n , $\alpha(n)$ becomes practically independent of n , resulting in a scaling behavior of $\alpha(n) \sim O(1)$.

If we are to study recursive partitioning, we must consider a sequence of circuit (sub)graphs for which:

$$\begin{cases} G_k &= 2^{H-k} \\ t'_k &= t_G - \frac{T_k}{G_k} \\ p_{E,k} &= \frac{(1/2)G_k t'_k}{(1/2)G_k(G_k-1)} \end{cases} \quad (7.28)$$

If we consider a circuit with a strictly Rentian (optimal) partitioning behavior ($T_k = t_G G_k^p$) as an approximation, this becomes:

$$\begin{cases} G_k &= 2^{H-k} \\ t'_k &= t_G (1 - G_k^{p-1}) \\ p_{E,k} &= t_G \frac{1 - G_k^{p-1}}{G_k - 1} \end{cases} \quad (7.29)$$

In the limiting cases for the Rent exponent p , we find $\lim_{p \rightarrow 0} p_{E,k} = t/G_k$ and $\lim_{p \rightarrow 1} p_{E,k} = 0$. To understand this second limit, remember that the edge probabilities only relate to *internal* edges. A strictly Rentian behavior with a Rent's exponent equal to one means that all edges are external. Hence, $p_{E,k} = 0$ and the number of terminals of a module is proportional to the module size.

However, as long as p is not too close to 1 and for sufficiently large module sizes, the sequence of (sub)graphs resulting from optimal bipartitioning has a node degree that is approximately independent of n and an edge probability scaling law of (approximately) $O(1/n)$ (Figure 7.8). Note that the $p_{E,k}$ in Equations (7.28) and (7.29) can not be interpreted strictly as probabilities, since they turn out larger than one for small module sizes and low values of p (see Figure 7.8). This expresses the fact that, to obtain strictly Rentian behavior with a given value of t_G , multiple edges must be present between module pairs at the lowest hierarchy levels. This indeed occurs frequently in those synthetic benchmarks in the \mathbf{q} - and \mathbf{s} -series (\mathbf{qXX} and \mathbf{sXX}) with the lowest Rent parameter values.

From this scaling behavior, we can conclude that our (sub)graph sequences probably correspond better to the class of *sparse random graphs*,

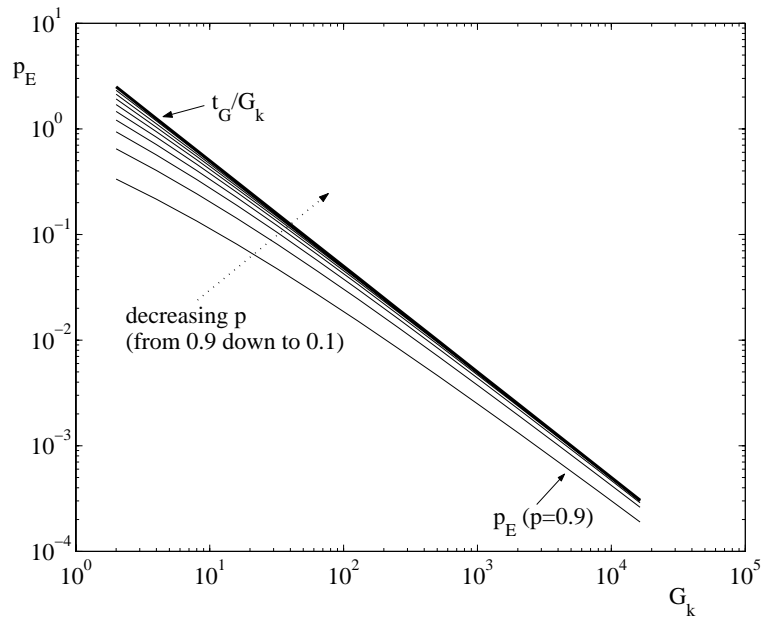


Figure 7.8: Scaling limits of the edge probability p_E in a sequence of subgraphs resulting from recursive circuit partitioning (for strictly Rentian circuit graphs with $t_G = 5$ and a range of values for p).

also studied by Fu and Anderson, having a constant mean node degree⁴ and an edge probability $p_E \sim 1/n$. Unfortunately, they had to conclude that in this case, finding an exact solution to the minimum partitioning problem is intractable. Since their paper, several efforts by other researchers have only led to the same conclusion.

In [SM99], Schreiber and Martin studied the bisection cut sizes of sparse random graphs in a more empirical fashion. Part of their motivation was the fact that in the very large and complex problems addressed in engineering applications, it is important to be able to predict the cut size \mathcal{C} resulting from the heuristic used for bipartitioning, rather than the exact theoretical optimal solution \mathcal{C}_0 . Since heuristics are usually based on some degree of randomization, Schreiber and Martin discuss the distribution of the optimized solutions found by different heuristics.

They find, from both (partial) theoretical arguments and experiments, that these distributions, across $\mathcal{G}(n, p_E)$ and a large number of experiments, tend towards Gaussians and that they become peaked for sufficiently large graphs. No rigorous proof is given for this statement. However, the authors of [SM99] argue that it is at least plausible and confirmed by their experimental data. They point out that the distribution of \mathcal{C} can be considered as the sum of a large number of (binary) random variables (expressing the event of each individual edge being cut or not) that are *not too correlated*. Since, according to the central limit theorem, the normalised distribution of the sum of a large number of random variables does converge towards the normal distribution, there is reason to believe that this might also (to some extent) be the case for \mathcal{C} .⁵ More importantly, their experiments indicate that both the mean and the standard deviation of \mathcal{C} show a $O(n)$ scaling behavior for large random graphs. Since the mean number of edges also scales $\sim n$, the cut probabilities for optimal bipartitionings of sparse random graphs also approach a constant value for large n . This is in accordance with the single parameter model for α_k proposed by Verplaetse (Equation(4.19) and Figure 4.7).

However, it does not correspond to the cut probabilities observed in optimized bipartitioning of circuit graphs. This again confirms the observation made in Section 7.2.3, that the essential topological properties

⁴Note that in their model, the node degree is itself still a random variable.

⁵Note that in [SP91], Sastry indicates that the distribution of \mathcal{C} , as well as the distributions of the cost functions occurring in a broad range of combinatorial optimization problems, follow a Weibull distribution, rather than a normal distribution.

of most real-world graphs in general and of circuit graphs in particular are fundamentally different from those of (uniform) random graphs.

In [Boe01], the meta-optimization heuristic *extremal optimization* (EO) is applied to optimal graph bipartitioning for dense and sparse random graphs as well as for different classes of sparse random graphs (i.e., with the average node degree independent of the graph size n) and non-uniform edge probabilities. These graphs were generated according to the following geometrically dictated graph generation strategies:

ferromagnetic graphs: the nodes are arranged according to a cubic lattice; a fraction x of all nearest neighbor node pairs is connected. The average node degree was chosen to be $6x = 2$ in the paper.

geometric graphs: the nodes are randomly distributed in the two-dimensional unit square; an edge is created between all node pairs that are no more than a distance d apart. The value of d was chosen to obtain an average node degree of 6.

For these classes of graphs, Boettcher reports a general scaling behavior of the optimized cut size $\mathcal{C} \sim n^{1/\nu}$, with $\nu = 1$ for both random graph types, in accordance with the other results we already mentioned. For the ferromagnetic graphs, he finds $1/\nu \approx 0.75 \pm 0.05$ and for the geometric graphs $1/\nu \approx 0.6 \pm 0.1$. The corresponding cut probability scaling behavior is given by $\alpha \sim n^{(1/\nu)-1}$, resulting in $\alpha \sim n^{-0.25 \pm 0.05}$ and $\alpha \sim n^{-0.4 \pm 0.1}$ for ferromagnetic and geometric graphs, respectively. Although these graph classes are definitely not circuit graphs, their scaling behavior is consistent with that found for recursive circuit partitioning in Chapter 4. Indeed, for large circuit modules, we find approximately $\alpha_k \sim G_k^{-\epsilon}$ for Verplaetse's model (4.20) and $\alpha_k \sim G_k^{-\beta}$ for our own power law model. The scaling exponent values for the ISPD98 benchmark circuits were found to be ranging from 0.43 to 0.62 for $-\epsilon$, and from 0.33 to 0.45 for $-\beta$.

Obviously, further research is required into the partitioning behavior of different classes of graphs and the link between graph topological properties and this behavior. However, the results discussed in this Section are in accordance with the models developed in Chapter 4. Furthermore, we might conjecture that the scaling trends for partitioning behavior might be similar for many classes of non-random graphs as was the case for other graph properties.

7.3 Conclusions

While the number of unsolved issues mentioned in this chapter may seem huge, we have good hopes of being able to solve many of them as a continuation of the present Ph.D. research. It is our belief that, at least for many issues described in the first section, the understanding and the results we have already obtained have moved those problems from almost insolvable to soon-to-be solved.

The more fundamental problems regarding a deeper understanding of the circuit topology will probably prove more difficult and would best be tackled by joining efforts with mathematicians, statistical physicists or people from the interdisciplinary network community in general.

Chapter 8

Conclusions

In this chapter, we summarize the most important conclusions and achievements of this Ph.D. thesis. They can be divided into two categories. On the one hand, the analytical Chapters 3 and 4 have led to a better understanding of fundamental approaches to predicting wire length distributions and the terminal-gate relationships they rely on. On the other hand, the resulting insights were applied in Chapters 5 and 6 to obtain more reliable a priori predictions of the wire-length distribution and the expected minimal clock cycle for design space exploration. To conclude our work, we have indicated the most important topics for further research as well as possible ways to address them.

8.1 Analysis and understanding

Our work targeted the prediction of interconnect-related cost metrics for the exploration of trade-offs between different design alternatives or technological variants, prior to physical design. From this point of view, the most important requirement is to obtain a good correlation between the prediction and the effective implementation across alternative implementation solutions. Additionally, a high accuracy of the predicted metrics would be welcome, but it is not strictly necessary.

From this perspective, we have analysed the two most popular fundamental approaches to the a priori prediction of interconnect properties, Donath's approach and Davis' approach (Chapter 3). This analysis was focused on the ability of those models to predict wire length distributions for individual circuits with two-terminal nets only, placed in gridded placement architectures with different grid and cell aspect ra-

tios. They were also evaluated based on the achieved correlation of the predicted average wire lengths with experimentally measured values.

For both prediction approaches, we found that the Rent exponent that is used has a very large impact on the quality and the reliability of the predicted values. From the descriptions of Donath's and Davis' prediction models, it was clear that both have a different perspective on how circuit modules are to be generated to extract the Rent exponent. In Donath's technique, a recursive partitioning of the circuit netlist is considered to be the basis for the Rent characteristic. Although this is a computation-intensive operation, only the circuit netlist is required for it. In contrast, in Davis' technique, modules are defined by "cutting" regions from a circuit layout, i.e. *after circuit placement*. This was identified to be an essential obstacle for the application of this technique to predict interconnect properties for individual circuits.

Additionally, our evaluation results indicated that the application of Rent's rule as an approximation for the Rent characteristic might in itself be limiting the achievable accuracy of the predictions.

Finally, whereas Davis' model is easily extendible to support the required architectural flexibility, this is far less obvious for Donath's prediction approach.

As terminal-gate relationships for circuit modules appear to be the key to good predictions, we have performed a thorough exploration of them. First, we ascertained by simple measurement that the Rent characteristics resulting from recursive partitioning and those obtained from layout regions (the average layout Rent characteristic) can be quite different.

Subsequently, we have studied both types of Rent characteristics. For the partitioning Rent characteristic, our discussion was focused on trying to reduce the computation power that is required to obtain them. We have discussed and improved Verplaetse's model for such Rent characteristics. The result is a numerical, single-parameter model (the β -model) that allows the estimation of the entire Rent characteristic by performing only a limited number of partitioning levels. In addition, it opens perspectives towards the abstraction of interconnect prediction models to even earlier stages in the design cycle, where the details of the circuit's netlist topology are not yet known. In Chapter 7, we found that both Verplaetse's model and its extensions by us are in accordance with graph partitioning results from other research domains.

For the average layout Rent characteristic, we also went in search

of better models than Rent's rule. We discussed Christie's differential equation as well as our own numerical variant. Although both result in terminal-gate relationships that are much closer to the average layout Rent characteristic than Rent's rule, they offer no indications of the relationship between them and properties of the graph topology that can be measured a priori. Since this was the main requirement for the applicability of Davis' model to predictions for individual circuits, we developed a new numerical model, based on the relationship between the distribution of placement wire lengths and the number of terminals for a layout region. Results from this model were found to accord very well with measured average layout Rent characteristics.

Our model was then applied to determine the theoretical bounds for average layout Rent exponents in two-dimensional placement grids: $p_{(al)} \in [0.5, 1]$. We were also able to establish that both the position and the shape of a region have a significant impact on its expected number of terminals.

We used these findings to explain why, even when a more accurate terminal-gate relationship is used in Davis' prediction model, the resulting wire length distributions still did not accord well with measured ones. Indeed, regions with very different shapes are used in this model and the same terminal-gate relationship is applied to all of them. This approximation was shown to be incorrect, both by theoretical and empirical arguments.

Finally, our enumeration model for average layout Rent characteristics shows these characteristics to be the consequence of the placement wire length distribution, rather than its cause. Hence, they cannot be determined a priori for individual circuits in order to predict their placement interconnect properties.

Overall, our analysis has led to the conclusion that a prediction technique based on Donath's model was by far the most promising candidate to achieve our goals.

8.2 Improvements to the prediction of interconnect properties

In Chapter 5, we have systematically alleviated the most important restrictions of Donath's model. Based on our findings regarding the importance of the terminal-gate relationship in this model, we have first

moved to a numerical model that uses the Rent characteristic instead of Rent's rule. This resulted in a drastic improvement of the correlation between predicted and measured average wire lengths (e.g., from 0.804 to 0.983, when compared to `Plato` placements). To support the necessary flexibility with respect to placement architectures, we provided an extension to incorporate homogeneously distributed empty space in our model and moved from strictly balanced four-way partitioning to two-way partitioning that is allowed to be slightly unbalanced. The result is a very flexible numerical model that obtains correlations of the average wire length for square grid placements ranging from 0.985, when compared to `Plato` to 0.952, when compared to very good placements with `SA`. The correlations for individual benchmarks across a wide range of architectural variants are similar. We can conclude that this model is at least very reliable in predicting and comparing average wire lengths.

However, although they show a good correlation, the predicted values are a very large over-estimation of the measured ones. To try and improve the accuracy of our model, we have explored two occupation probability functions. The first, proposed by Stroobandt, results in a large under-estimation, even compared to the `SA` placement. The second, however, which was proposed by Verplaetse, resulted in predicted average wire lengths that were on average only 6.75% above those obtained by `Plato`. Also, the predicted wire length distributions very closely match the measured distributions. This close match is most pronounced for the tail of the distribution.

Finally, we have investigated the sensitivity of our model to variations in the Rent characteristic. It was found to be relatively insensitive to variations due to the randomized nature of the partitioning tool that is used. However, the deviations that occur when estimating the Rent characteristic from a limited number of partitioning levels (using the β -model proposed in Chapter 4) can be quite large. Still, with our β -model, the accuracy and reliability of our predictions remain much better than when Rent's rule is used.

In Chapter 6, we have considerably improved existing techniques for the *a priori* prediction of the minimal clock cycle in synchronous circuits. This was achieved by combining the wire length distribution with an approximated delay equivalent graph topology with statistically independent paths, for which the *maximum* path delay can be easily calculated. Our new model was first evaluated by using the mea-

sured wire length distribution as an input. This resulted in correlations between measured and predicted average minimal clock cycles of 0.97 (compared to 0.95 for the existing models). On average, our predictions were between 8.5% and 10% below the measured values (depending on the technological parameters used for estimating the delays), whereas the traditional approach yielded average underestimations ranging from 24% to 27%. By performing a Monte Carlo simulation that ensures that the homogeneity assumption is fulfilled, we were able to assess that our predictions are extremely accurate when its underlying assumptions are met. From this experiment, we can conclude that the fact that the homogeneity assumption is not always fulfilled is the most probable cause of most of the remaining errors in our model.

In combination with predicted wire length distributions, the reliability of our model does not decrease drastically (the correlations decrease from 0.97 to 0.967). However, even the smallest systematic error on the predicted wire length distributions immediately affects the accuracy, and we find an average over-estimation between 13.5% and 14.5%.

8.3 Future work

In general, we can conclude from the previous section that with our new prediction models for the wire length distribution and even for a metric as complex as minimal clock cycle, reliable predictions of interconnect properties for individual circuits are now becoming feasible. However, in Chapter 7 many remaining issues were identified.

Within the framework of assumptions of this Ph.D. thesis, the prediction accuracy can still be increased by providing occupation probability models that can be tuned to the performance of specific tool flows. Also, to further improve our clock cycle predictions, local deviations from the homogeneity assumption need to be modeled. However, the extensions of this framework are even more pressing. In particular the extension to include multi-terminal nets will be one of our first subjects for further investigation. We are confident that, with our model extensions and the understanding reached in this work, a solution is within reach. Another priority is the extension towards different types of inhomogeneity, which is required to achieve more accurate predictions of interconnect-related cost metrics such as achievable clock cycle, power consumption or routability.

More fundamental and long-term research should be directed at a more accurate modeling of circuit graph topologies. A start to this research was made in Verplaetse's Ph.D. thesis [Ver03] and we, too, have suggested several lines of inquiry on this topic in Chapter 7. However, the problems in this area would probably better be tackled in cooperation with scientists from other research domains such as graph theory and statistical mechanics.

Appendix A

Experimental setup

Since the main topic of this PhD-thesis is the prediction of interconnect properties, and in particular of placement wire lengths, we have performed a large number of placement experiments to illustrate and validate the models presented. In this appendix, we give an overview of the benchmark circuits we used for those experiments and their main properties. We also discuss the placement tools and summarize the placement results.

A.1 Benchmark circuits

A.1.1 Synthetic benchmarks

Interconnect prediction models are often based on a cascade of assumptions and simplifications. In some cases it is useful to try and find out the errors introduced by each step, or to validate a model when all of its underlying assumptions are met. That is why we have performed many of our experiments on synthetic benchmark circuits, generated by our in-house benchmark generator `gnl` [SVVC00]. Hence we were able to create several sets of benchmark circuits with controlled size G_c , number of external terminals T_{ext} , (average) number of terminals per gate t_G and complexity of their interconnect topology (i.e., their partitioning Rent characteristic - Section 2.2).

The way in which `gnl` generates benchmark circuits is by recursive bi-clustering. This results in a *clustering tree* or, when traversed from top to bottom, a *recursive partitioning tree* for each circuit. Between each pair of clusters, several connections are made, with their

number derived from the target (partitioning) Rent characteristic. Note that, when repartitioning the resulting circuits, the resulting partitioning Rent characteristic may deviate slightly from the generated characteristic. To be able to reproduce the generated terminal-gate relationship in, e.g., a recursive partitioning based placement process, the generated partitioning trees were saved.

The generation process can be controlled by many different options and parameters. For instance, the net degree distribution can be dictated to some extent or circuits can be generated with two-terminal nets only. Also, a library of gate types can be provided, for each type defining the number of input and output terminals and its frequency of occurring in the circuit. In general, many parameters can be made either deterministic, or probabilistic in such a way that only their first and second moments are controlled.

For our experiments, we have used only very few of the options provided in `gnl`. Our aim was to create very regular and homogeneous circuits, with properties as close as possible to those assumed in the interconnect prediction models.

We have used two series of 17 synthetic benchmark circuits of $G_c = 16384$ gates each and four terminals per gate *exactly* and two-terminal nets only.

The \mathfrak{q} -benchmark series: The 17 benchmarks in the first series were generated with increasing values of the Rent exponent, ranging from 0.10 to 0.90 with increments of 0.05, and with an almost perfect adherence (at generation) to Rent's rule. This means that these circuits show neither a region II nor a region III in their generation Rent characteristics. They were named $\mathfrak{q}10$ to $\mathfrak{q}90$, reflecting their respective generation Rent exponents.

The \mathfrak{s} -benchmark series¹: A second series of 17 benchmarks was generated, only differing from the first one in that their Rent characteristics do show a region II (which was also enforced), and hence they correspond somewhat better to real circuits. They are named $\mathfrak{s}10$ to $\mathfrak{s}90$ (again, reflecting their generation Rent exponent values).

Because the enforced Rent characteristics determine the number of

¹The prefixes \mathfrak{q} and \mathfrak{s} used for naming the benchmarks have no particular meaning. Our benchmarks are just two series amongst a number of other generated series, each with a different letter as a prefix.

p	Benchmark	T_{ext}	N_c	Benchmark	T_{ext}	N_c
0.10	q10	10	32763	s10	8	32764
0.15	q15	18	32759	s15	10	32763
0.20	q20	28	32754	s20	14	32761
0.25	q25	46	32745	s25	20	32758
0.30	q30	74	32731	s30	26	32755
0.35	q35	120	32708	s35	36	32750
0.40	q40	194	32671	s40	50	32743
0.45	q45	316	32610	s45	68	32734
0.50	q50	512	32512	s50	94	32721
0.55	q55	832	32352	s55	128	32704
0.60	q60	1352	32092	s60	176	32680
0.65	q65	2194	31671	s65	242	32647
0.70	q70	3566	30985	s70	330	32603
0.75	q75	5792	29872	s75	452	32542
0.80	q80	9410	28063	s80	620	32458
0.85	q85	15286	25125	s85	850	32343
0.90	q90	24834	20351	s90	1166	32185

Table A.1: Circuit parameters for the benchmarks of the q and s series. All of these benchmarks have $G_c = 16384$, $t_G = 4$ and two-terminal nets only.

terminals per gate t_G and the number of external terminals T_{ext} in the circuit, they also affect the number of internal nets N_c :

$$N_c = \frac{t_G G_c - T_{ext}}{2} \quad (\text{A.1})$$

Table A.1 summarizes all relevant circuit parameters for both the q and the s benchmark series. Figure A.1 shows the generated partitioning Rent characteristic for a representative of each series (q65 and s65, respectively).

A.1.2 Public domain benchmark circuits

Besides the synthetic benchmark circuits, we have also performed experiments on a large number of benchmarks that are generally assumed to represent (somewhat) *realistic* circuits. These benchmarks, which are publicly available, are often used in literature to provide reproducible results. Hence, they are considered as a sort of standard reference set for reporting, e.g., CAD and prediction results. Again, to be able to perform the necessary experiments, we needed to use two different benchmark series: the ISPD98 benchmarks and the MCNC benchmarks.

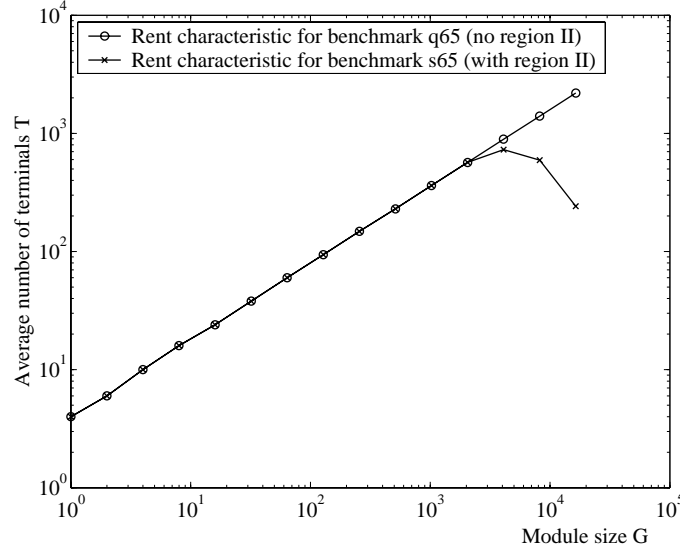


Figure A.1: Partitioning Rent characteristic for two circuits with $p = 0.65$, one without region II (q65) and one with region II (s65)

The ISPD98 benchmarks

For most of our experiments, we use the ISPD98 industrial benchmark suite [Alp98]. These 18 benchmarks, named `ibm01` to `ibm18`, are freely available at and are frequently used for the validation of CAD-tools as well as interconnect prediction techniques. They have sizes ranging from $G_c = 12506$ to $G_c = 210341$ and are currently the largest that are publicly available. Because in this work we concentrate on models that do not include multi-terminal nets, we have created a two-terminal net version from these benchmarks. These were generated by replacing all multi-terminal nets by multiple two-terminal nets with the same source. Note that throughout this PhD thesis, when referring to any of the benchmarks `ibm01-ibm18`, these two-terminal net versions are implied. The properties of these benchmarks are summarized in Table A.2.

The MCNC benchmarks

In Chapter 6, we look into the possibility of predicting the minimal clock cycle from the wire length distribution. To validate our prediction approach, we needed circuit netlists that not only contain signal

Benchmark	Internal gates (G_c)	Average terminals per gate (t_G)	Internal nets (N_c)	External nets (T_{ext})
ibm01	12506	5.81	36207	246
ibm02	19342	6.36	61359	259
ibm03	22853	5.78	65893	283
ibm04	27219	5.37	72940	287
ibm05	28146	6.91	96659	1201
ibm06	32332	5.77	93195	166
ibm07	45639	5.58	127235	287
ibm08	51022	5.99	152643	286
ibm09	53110	6.07	160914	285
ibm10	68685	6.46	221618	744
ibm11	70152	5.68	198924	406
ibm12	70439	6.82	239879	637
ibm13	83709	6.14	256910	490
ibm14	147088	5.35	393497	517
ibm15	161185	6.55	527690	383
ibm16	182980	6.43	588304	504
ibm17	184752	7.25	669725	743
ibm18	210341	5.87	617531	272

Table A.2: Circuit parameters for the ISPD98 benchmarks, measured after the conversion of multi-terminal nets to two-terminal nets.

direction, but also synchronization information, i.e., the positions of flipflops, or an indication of which net sources can be assumed to carry signals that are synchronized with the circuit clock signal. Unfortunately, only the circuit topology is available for ISPD98 benchmarks. In particular, they contain no synchronization information, so they can not be used to evaluate models for clock cycle prediction. The only publicly available benchmarks that are suitable for this purpose are the *Logic Synthesis benchmarks* from the NCSU Collaborative Benchmarking Laboratory CBL (<http://www.cbl.ncsu.edu/>). However, these benchmarks date from 1995 and before. Several series of benchmarks are available on their website, but many of those consist (at least in part) of benchmarks with the same name. We have selected the most recent of those series, i.e., the benchmarks in the LGSynth95 series, which are available as logic netlists. Because of their age, even the largest of these benchmarks are relatively small and the smallest are almost trivial. From these circuits, we have retained only those with more than 500 gates, for which we again replaced all multi-terminal nets by multiple two-terminal nets. This leaves a total of 68 benchmarks, with sizes ranging from 527 to 24819 gates and maximum logic depths ranging from 6 to 284. The relevant circuit parameters for these circuits are summarized in Tables A.3, A.4 and A.5. The majority of these benchmarks (Tables A.3 and A.4) are combinational and hence contain no flipflops. For the ones that are sequential, we have also mentioned the number of flipflops (Table A.5).

A.2 Placement tools

As was the case for our choice of benchmarks, we have chosen different types of placement approaches. On the one hand, we wanted to compare predicted values to experimental placement results that were generated in a way as close to the prediction model as possible, while on the other hand, we wanted to compare with industrial-style placement results, as well as with very good placements that serve as an approximation to the optimal placement.

To be able to perform circuit placements under carefully controlled conditions, we have used our own placement tool `Plato` [VDSVC01]. This tool supports many different options, all within the framework of recursive partitioning based placement. The industrial placements were done with the commercial tool `Qplace` (version 5.0.55, incorpo-

Benchmark	Internal gates	Internal nets	External nets	Maximum logic depth
pdcc	7609	20161	56	15
spla	7534	19957	62	15
ex1010	5386	14952	20	15
C6288	4896	7184	64	248
C7552	4722	6482	314	69
misex3	3680	9845	28	13
C5315	3512	4993	301	81
apex2	3233	8302	42	13
i10	3157	5222	481	57
seq	3027	7571	76	12
des	3009	8980	501	12
apex5	2829	6173	205	11
pair	2825	4165	310	36
daluc	2716	4482	91	61
alu4	2425	6607	22	13
frg2	2187	3908	282	22
cordic	2106	4948	25	14
C2670	1828	1952	297	48
C3540	1814	2768	72	68
x3	1738	2844	234	26
ex4p	1710	3672	156	11
ex5p	1541	4194	71	12
cps	1461	3093	133	11
apex4	1450	4123	28	11
i8	1385	4406	214	11
apex3	1246	3002	104	12
apex1	1179	2824	90	12
too_large	1155	15566	41	6
t481	1093	2658	17	13
apex6	1090	1386	234	16
rot	1080	1970	242	22
C1355	1054	1458	73	46

Table A.3: Circuit parameters for the combinational MCNC benchmarks (partim: benchmarks with more than 1000 internal gates), measured after the conversion of multi-terminal nets to two-terminal nets.

Benchmark	Internal gates	Internal nets	External nets	Maximum logic depth
i7	934	1374	266	6
misex3c	906	2264	28	11
C1908	881	1238	58	57
x4	839	1436	165	11
b12	802	1854	24	11
C880	791	983	86	42
e64	783	1434	130	8
table3	782	1963	28	12
table5	765	1896	32	12
i6	750	1100	205	6
term1	687	1420	44	24
C499	670	970	73	28
i9	630	1458	151	10
i5	597	622	199	9
x1	537	2508	86	6
example2	535	640	151	15
rd84	511	1435	12	11

Table A.4: Circuit parameters for the combinational MCNC benchmarks (partim: benchmarks with less than 1000 internal gates), measured after the conversion of multi-terminal nets to two-terminal nets.

Benchmark	Internal gates	Internal nets	External nets	Flip- flops	Maximum logic depth
clma	25622	48604	465	66	79
s38417	24255	33275	135	2930	66
s38584	20349	32793	343	2852	71
dsip	4079	5951	426	448	22
bigkey	3661	11487	460	448	10
s9234	3083	4078	76	270	56
s5378	3076	4341	85	326	34
parker1986	2795	4369	59	356	62
mm30a	2059	3747	64	180	140
daio_receiver	1942	3464	63	166	46
phase_decoder	1671	3234	14	110	34
ecc	1618	2481	26	230	20
sbc	1147	1666	97	54	23
gcd	1114	1750	44	118	34
s1423	916	1350	23	148	68
s298	880	7185	10	16	6
mm9b	777	1355	22	52	81
s1238	766	1254	29	36	33
s1196	763	1198	29	36	35
scf	750	3047	84	14	6
s953	730	1020	40	58	28
s838	665	870	38	64	95
mm9a	631	1122	22	54	56
mult32a	565	1064	35	64	76
s713	515	609	59	38	87

Table A.5: Circuit parameters for the sequential MCNC benchmarks, measured after the conversion of multi-terminal nets to two-terminal nets.

rated in Cadence Silicon Ensemble DSM version 5.2). This tool is optimized to perform relatively good placements within a very limited time. Finally, we have implemented a *flat*, homogeneous placement using simulated annealing. While this tool performs much better placements than any of the other tools, it takes unreasonably long computation times to achieve this (up to several months for the largest ISPD98 benchmarks on a 1 GHz PC with sufficient RAM to avoid swapping).

A.2.1 Recursive partitioning based placement

Placement using `Plato`

The first step in recursive partitioning based placement in `Plato` is the creation of a bipartitioning tree by recursively applying the `hMetis` partitioning tool, configured for minimal imbalance between the module sizes resulting from each cut. Alternatively, a partitioning tree can be read from file, to allow, e.g., the use of the partitioning trees generated by `gn1` for the synthetic benchmarks.

To perform global placement, a partitioning schedule is then created for the placement grid. As stated in Chapter 2, we allow for placement grids with different numbers of grid positions as well as different unit distances in each direction. The grid partitioning schedule performs *balanced* hierarchical two-way partitioning, at each step choosing the cut direction that minimizes the maximal wire length in the resulting modules. If the grid side that is cut consists of an odd number of grid locations, one row or column is added to ensure that the partitioning can be balanced.

Finally, since `hMetis` does not allow the enforcement of perfectly balanced partitions, it is possible, due to an accumulation of slightly larger modules in one branch of the partitioning tree, that not all branches in the circuit partitioning tree have equal depth. In this case, “empty” gates are added to the less populated branches and all leaves are pushed down to the final common tree depth.

It is also possible that the resulting partitioning tree has a larger tree depth than the grid partitioning tree. In this case, the depths of both trees are further extended until they match, with the restriction that the depth of the grid partitioning tree can only be extended by two orthogonal cuts at the time. Hence, if this depth must be increased, every grid location is replaced by a square of 2×2 grid locations, effectively dou-

bling the X and Y dimensions of the grid. This is done not to disturb the grid aspect ratio, since this relates to the aspect ratio of the target placement grid.

The final global placement grid is often larger than the original one, and its number of grid positions in each direction is always a power of two. When both tree depths match, the modules in the circuit partitioning tree are randomly assigned to the grid modules, following the grid partitioning schedule. Note that the synthetic benchmarks in the q - and s -series were generated according to a perfectly balanced clustering tree. When using this tree for hierarchical partitioning based placement in a grid with exactly 16384 gate locations (i.e., as many as there are gates in the circuits) and for which X_g and Y_g equal powers of two, the depths of both the circuit and the grid partitioning trees are equal to 14 and the global placement grid contains no empty grid locations.

After the initial global placement is created, a swapping step can be applied. In this swapping, the partitioning tree remains unchanged: only the assignment of that tree to the grid partitioning tree is explored.

To validate Donath-based interconnect prediction techniques, we have provided the following random swapping scheme:

Random quad-swapping: for a sequence of two orthogonal cuts, the random bi-swapping schedule does not result in equal probabilities for each of the 24 possible assignments of four circuit modules to the four module positions; in the quad-random scheme, the probabilities are adapted to make these assignments equally probable, enabling an evaluation of Donath's original model.

The placement heuristic that consists only of performing a recursive partitioning and a random quad-swapping assignment of the modules will be referred to as `Plato-Quad`.

For all other purposes, i.e., to create well-optimized placements, the partitioning tree is traversed top-down several times, at each step taking a greedy swapping decision.

The global placement stage results in gate locations that are more or less optimized relative to each other, i.e., a two-dimensional ordering of the gates, in a grid with similar aspect ratio to the target grid. To obtain the final placement, a detailed placement stage follows. If the global placement grid is larger than the target placement grid, it is first compacted. In an ideal situation, this would result in a reduction of

the total wire length which is proportional to the square root of the area reduction. However, because overlapping gates are not allowed, a valid compaction usually requires the (optimized) relative ordering of the gates to be disturbed. Our compaction is done in several steps, according to a strategy that was developed to maintain the relative gate positions as much as possible. We achieve this by trying to identify those regions for which compaction can be performed at low cost, while leaving the “expensive” regions unchanged when possible.

The core of our compaction algorithm is again based on recursive partitioning. The source placement grid is mapped onto a smaller target grid, by recursive partitioning of both grids, at each levels assigning the source grid modules to the target grid modules. At each hierarchy level, at least one grid dimension is cut. This cut is chosen in such a way that:

1. the total number of occupied gates for each source grid module does not exceed the number of available positions in the target grid module;
2. each dimension of the source grid module which needs to be compacted is cut, without violating constraint 1;
3. the partitioning is as balanced as possible without violating constraint 1;
4. if, within the previous constraints, some freedom exists in assigning the source grid modules to the target grid modules, the source module for which most positions are occupied is assigned to the largest target module and vice versa.

The last of those rules ensures that the most dense regions will be disturbed as little as possible.

Our approach is illustrated in Figure A.2. In the first step, the reader can verify that the bottom right source module, which has most of its grid positions occupied, is mapped onto the largest target module. Whenever a source and target module pair have equal dimensions, they are *terminated*, i.e., the gates from the source module are assigned to the same positions in the target module (relative within the module). In our example, this situation occurs for the first time in step 2, where three modules have reached their final dimensions. Modules that are still too large are first checked for empty rows or columns. These can be

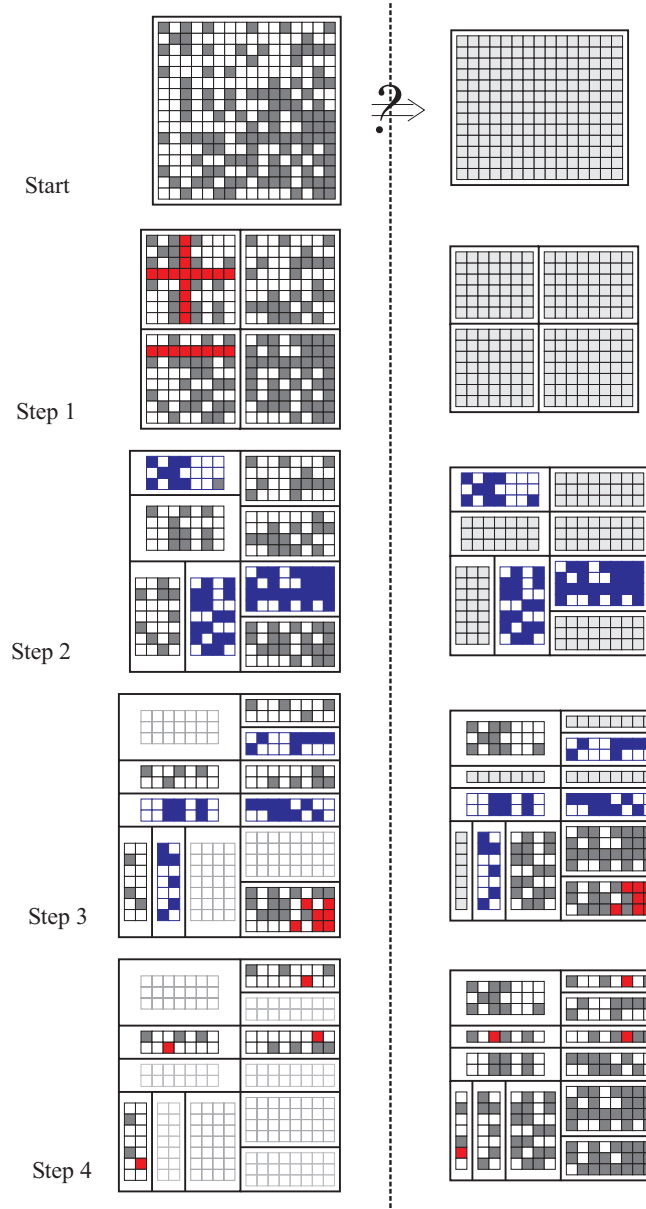


Figure A.2: Illustration of our placement grid compaction, based on recursive grid partitioning. Black squares represent occupied grid positions, white squares are empty. Each step shows the partitioning of source and target grid modules. In step 1, rows and columns are eliminated (shown in red). In step 2, the first three modules are terminated (blue). In step 3, the squeeze algorithm is used for the first time for the lower right module. For each module that is squeezed, shifted gates are marked in red.

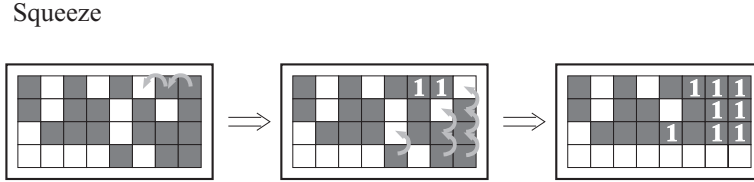


Figure A.3: Illustration of our squeeze algorithm, applied to reduce the number of rows from four to three. In the rightmost column, there is no empty position, so the nearest column with an *extra* empty position is identified (column 6). This empty position is moved to the last column by shifting gates to the left. Then, the bottom row is emptied by shifting gates upwards. The numbers in the grid position indicate the total (Manhattan) distance they have been moved.

eliminated without changing the relative ordering of the gates *within the module*. In step 1 of our example, this operation is used to very cheaply reduce the dimensions of the two source modules on the left. Any modules that are still too large are further partitioned.

If, for a given source and target grid module pair, a valid partitioning can not be found, the source module is *squeezed* into the target module. The *squeeze* operation compacts the grid by one row or column at the time, alternating between rows and columns, at the cost of disturbing the relative ordering of the gates. This happens, e.g., for the lower right module in step 3 of our example, where the number of rows needs to be reduced from four to three. As is illustrated in Figure A.3, an empty position is first created in each column (row) that does not have one, by shifting the gates a minimum number of positions. Then, the bottom row (rightmost column) is emptied by shifting gates upward (to the left) and eliminated from the grid. Whenever multiple possibilities exist, a choice is made to minimize the number of gates that needs to be shifted and the distance they are shifted.

Because the compaction can never be perfect with regard to maintaining the relative gate positions, a final greedy optimization is performed that consists of repeatedly traversing all gate pairs and swapping them if that improves the global cost function.

Placement using Qplace

The placement performed by Qplace also uses a recursive partitioning and assignment to obtain a global placement. However, in this tool a

slicing approach is followed, i.e., the partitionings must not be strictly balanced at every level. To optimize its placements, Qplace uses a mixture of greedy and hill-climbing optimization strategies, including swapping and simulated annealing.

In contrast to our own tool, a random “seed” can not be set in Qplace. This means that the variability on placement results for the same circuit can not be explored, since the same netlist and architecture settings will always result in the same placement result. An exploration of different solutions could be obtained by randomizing the net names and the order of the nets in the netlist file before placement, but due to lack of time, we have not done this.

However, we can expect the variability on the placement results to be rather small. Indeed, usually, the closer the placement results from a randomized tool are to the optimum, the smaller their variability can be expected to be. Since the results obtained by Qplace are systematically as good or better than those by Plato, and since they are obtained by a similar strategy, we can expect the variability to be no larger than that observed for Plato.

A.2.2 Flat placement by simulated annealing

Simulated annealing (SA) is one of the most frequently-used general-purpose hill-climbing algorithms for combinatorial optimization. It originates from the statistical mechanical model for the annealing process and has been thoroughly studied in literature (e.g., [OV89]). The purpose of is to find a *state* of a given system with minimal *energy*, starting from an initial state.

The inner loop of the procedure is called the *metropolis loop*. At each step, a new state is generated from the previous state according to certain generation rules. If the new state is better (i.e., has lower energy), it is maintained. If it has higher energy, it is accepted with a probability

$$p = \exp\left(-\frac{\Delta E}{kT}\right), \quad (\text{A.2})$$

where k is the Boltzmann constant, ΔE is the energy increase and T is a *temperature*. For very high temperatures, almost any new state will be accepted. The lower the temperature, the lower the probability of accepting a state which is worse than the current one. After a sufficient number of iterations of the metropolis loop is performed at the same

temperature, the system is said to be in *equilibrium*, and the probability of being in a state with energy e_i follows the *Boltzmann* probability density function:

$$P(E = e_i) = \frac{\exp(-e_i/kT)}{\sum_j \exp(-e_j/kT)}. \quad (\text{A.3})$$

If the system starts from a temperature value where almost every state is equally probable (the *liquid state* in the original annealing process) and the temperature is lowered so slowly that the system remains in quasi equilibrium, it will converge asymptotically towards its minimum energy state. However, if the cooling happens too quickly, the quality of the final result can be much worse. Therefore, an *annealing schedule* must be designed with care: it needs to be decided how many metropolis iterations at the same temperature have to be performed and how fast the temperature can be decreased. We have based our annealing schedule on that used in , e.g., [BR97].

When used as a homogeneous circuit placement heuristic, each possible placement is a state. The energy of that state is the value of the cost function that needs to be optimized, in our case the total wire length. Transitions from one state to the next are performed by randomly selecting a circuit gate and a new position for that gate. If the target position is empty, the gate is moved. If another gate was occupying that position, both gates are swapped. Because a very slow annealing schedule would take far too much time in commercial placement applications, several faster annealing schedules have been developed that perform relatively well, but not optimal. The speed-up is obtained by, e.g., cooling down very slowly only in a limited temperature range which is assumed to be the most critical (used in, e.g., [BR97]).

For our applications, we wanted to obtain really good placements, without caring much for computation time. Hence, we have chosen for a very slow cooling schedule for which:

- the initial temperature is chosen such that essentially all transitions are accepted;
- each temperature is maintained until equilibrium is reached (expressed as a sufficient stability of the average cost across the past states);
- when equilibrium is reached, the temperature is decreased by a factor of 0.999;

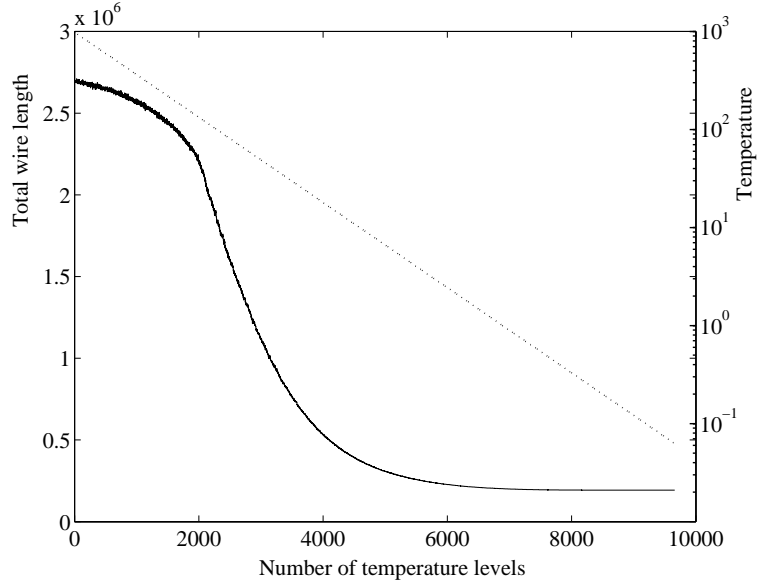


Figure A.4: Evolution of the total wire length (full line) and temperature (dotted line) as a function of the number of temperature levels for the SA-placement of benchmark `ibm01` in a square grid architecture. Temperature levels are a factor 0.999 apart.

- the annealing process is stopped when $T \leq 0.001E/N_c$, expressing that too few transitions are accepted to go on.

The resulting placements are significantly better than the ones obtained with either `Plato` or `Qplace`, but they did take up to several months on a 1GHz processor with sufficient memory to avoid swapping. Figure A.4 illustrates the evolution of total wire length and temperature for a SA placement of benchmark `ibm01`.

A.3 Extraction of Rent characteristics and Rent parameters

A.3.1 Measuring Rent characteristics

In this Ph.D. thesis, any measured Rent characteristic is a set of data points (G_i, T_i) which reflects the *average* terminal-gate relationship for a family of modules. For the different types of Rent characteristics, the

module generation strategies are described below.

Partitioning Rent characteristics: all partitioning Rent characteristics used in our experiments were obtained by recursively applying bipartitioning with `hMetis` [KK98]. At each partitioning step the imbalance was kept as small as possible with this tool. Note that it is not possible to enforce strictly balanced partitionings in `hMetis`;

Geometric Rent characteristics: for measuring average layout Rent characteristics, approximately 1000 randomly selected module positions were selected for each module size; these positions were always aligned with the placement grid.

For all types of Rent characteristic, the module sizes (number of gates in the module) were assigned to buckets, using the arithmetic average of the module sizes in each bucket as its G_i -value. An arithmetic average was also used to obtain the average number of terminals T_i for each bucket.

A.3.2 Fitting the Rent parameters

The Rent characteristic and its power law approximation

$$T = tG^p, \quad (\text{A.4})$$

known as Rent's rule, are omnipresent in interconnect prediction in general and in this work in particular. Since the fitted value of the Rent exponent p has a significant impact on the result of traditional interconnect prediction models (see Chapter 3), it is important to give an indication of the fitting procedure when reporting such estimations. This is also the case when the Rent exponents resulting from different Rent characteristics are compared (Chapter 4).

In our work, we use an automated fitting procedure for extracting the Rent parameters t and p from a Rent characteristic. In this way, a standardized and reproducible approach is used for all benchmarks, without any manual interaction or “tweaking”. The most important parameter in interconnect prediction is the Rent exponent p . Theoretical arguments and experimental results have shown that it is closely related to the exponentially decreasing part of the wire length distribution, largely independent of the placement tool (Chapter 3). Therefore,

Initial fit:**Select midpoint** R_a closest to (logarithmic) center of data range**Set region I** = $\{R_{a-1}, R_a, R_{a+1}\}$ **Calculate best linear fit on region I:** $T_{a,i} = t_a G_i^{p_a}$ **Second fit:****Calculate errors:** $e_i = \text{abs}(T_{a,i} - T_i)$ **Set region I** = $\{R_i : e_i < \mathcal{M}_1\}$ **Select midpoint** R_b closest to (logarithmic) center of region I**Set region I** = $\{R_{b-1}, R_b, R_{b+1}\}$ **Calculate best linear fit on region I:** $T_{b,i} = t_b G_i^{p_b}$ **Final fit:****Calculate errors:** $e_k = \text{abs}(T_{b,i} - T_i)$ **Set region I** = $\{R_k : e_k < \mathcal{M}_2\}$ **Calculate best linear fit on region I:** $T_{f,i} = t_f G_i^{p_f}$ **Figure A.5:** Automated fitting procedure summary

our procedure was optimized for fitting p , rather than for approximating the Rent characteristic. Hence, the resulting t -values can only be interpreted as scaling constants.

In the past, the region of the Rent characteristic that can be approximated by a power law has always been referred to as *region I* of Rent's rule. Obviously, when fitting a power law to region I, the essential part in such a procedure is the selection of data points that belong to it. The basic concept of our procedure is the intuitive notion that region I usually includes the central range (on a logarithmic scale) of module sizes in the Rent characteristic. Since our procedure operates entirely in the logarithmic domain, all comparative indications, as well as the best fits in the description below should be interpreted logarithmically. Using the notation $R_1 = (G_1, T_1), \dots, R_n = (G_n, T_n)$ for the data points in the Rent characteristic, the procedure is summarized in Figure A.5.

In this procedure, the first fit is based on the assumption that the central point is in the middle of region I. From this fit, a new estimation for the middle of region I is obtained. This results in a new fit, that serves to select the final region I data points, on which the final region I fit is based. The values of the error margins \mathcal{M}_1 and \mathcal{M}_2 were empirically optimized. They are set to 10% and 2%, respectively. Figure A.6 shows, for benchmark ibm07, the partitioning Rent characteristic, the initial and corrected midpoints R_a and R_b , the final selection of data

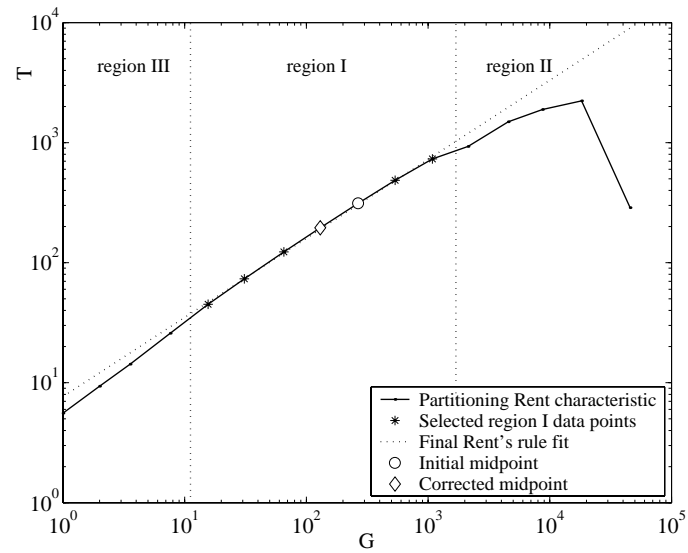


Figure A.6: Illustration of the automated procedure for identifying region I and fitting Rent's rule (using the partitioning Rent characteristic of circuit *ibm07*).

points belonging to region I and the Rent rule fit.

Appendix B

Basic probabilistic concepts

In this appendix, we briefly review some concepts from (discrete) probability theory that will be used or referred to in the context of this Ph.D. thesis. Although some of these concepts are probably well known to the reader, we mention them because they constitute the foundations on which probabilistic interconnect prediction is built.

B.1 Discrete random variables

The probabilistic techniques discussed in this work are all based on *discrete random variables*. In general, these are random variables X that can only take values in some countable subset x_1, x_2, \dots of real numbers. In the context of interconnect prediction, these variables are either confined to the integer domain or they originate from real-valued random variables X that have been assigned to equally spaced and non-overlapping *bins* or *categories* with central values x_i and width w such that:

$$X(Z) = x_i \Leftrightarrow x_i - w/2 \leq Z < x_i + w/2. \quad (\text{B.1})$$

Discrete random variables are characterized by their *probability mass function (pmf)* $f_X(x_i)$, which gives the probability of the random variable having a specific value x_i . It is given by:

$$f_X(x_i) = P(X = x_i), x_i = X_{min}, \dots, X_{max}, \quad (\text{B.2})$$

with

$$\sum_{x_i=X_{min}}^{X_{max}} P(X = x_i) = 1. \quad (\text{B.3})$$

This last condition expresses the fact that the pmf must always be *normalized* on its *domain* $X_{min} \dots X_{max}$. Although in general, $X_{min} \geq -\infty$ and $X_{max} \leq \infty$, the random variables considered in this work are all positive-valued and can take only a finite number of values.

The *cumulative distribution function* or *cdf* $F_X(x_i)$ of a discrete random variable is defined as:

$$F_X(x_i) = P(X \leq x_i) = \sum_{x_j=X_{min}}^{x_i} f_X(x_j). \quad (\text{B.4})$$

When only the cdf is given, the pmf can be derived from it by difference:

$$f_X(x_i) = F_X(x_i) - F_X(x_{i-1}). \quad (\text{B.5})$$

Two discrete random variables X and Y are said to be *independent* if

$$\begin{aligned} P(X \leq x_i, Y \leq y_i) &= P(X \leq x_i)P(Y \leq y_i) \\ &= F_X(x_i)F_Y(y_i) \end{aligned} \quad (\text{B.6})$$

and, as a consequence, also:

$$\begin{aligned} P(X = x_i, Y = y_i) &= P(X = x_i)P(Y = y_i) \\ &= f_X(x_i)f_Y(y_i). \end{aligned} \quad (\text{B.7})$$

If X and Y are *not* independent, then

$$\begin{aligned} P(X = x_i, Y = y_i) &= P(X = x_i)P(Y = y_i|X = x_i) \\ &= f_X(x_i)f_Y(y_i|X = x_i), \end{aligned} \quad (\text{B.8})$$

with $f_Y(y_i|X = x_i) \neq f_Y(y_i)$ the *conditional probability mass function* of Y , given that $X = x_i$ (for which it is assumed that $P(X = x_i) > 0$).

The *expected value* or *expectation* of a discrete random variable X is given by:

$$E[X] = \sum_{x_i=X_{min}}^{X_{max}} x_i f_X(x_i). \quad (\text{B.9})$$

Similarly, if it exists, the expectation of a function $g(X)$ of X is

$$E[g(X)] = \sum_{x_i=X_{min}}^{X_{max}} g(x_i) f_X(x_i). \quad (\text{B.10})$$

B.2 Moments and generating functions

The *moments* of a distribution are a special class of expected values. The k th general moment (usually simply referred to as *the* k th moment) is the expected value of X^k . The first moment, the expectation of X , is usually represented by the symbol μ_X . The expected value $E[(X - \mu_X)^k]$ is called the k th *central moment* of $f_X(x_i)$. By definition, the first central moment is always equal to zero, while the second central moment is usually referred to as the variance σ_X^2 and its square root is called the standard deviation σ_X .

The *moment-generating function* (mgf) $M_X(t)$ of X is the expected value of e^{tX} :

$$M_X(t) = E[e^{tX}] = \sum_{x_i=X_{min}}^{X_{max}} e^{tx_i} f_X(x_i). \quad (\text{B.11})$$

If it exists (which is always the case for discrete random variables on a bounded domain), the moment-generating function uniquely defines the pmf and cdf. Also, it has some very attractive properties, some of which will be discussed in the next section. $M_X(t)$ is called the mgf, because its k th derivative, evaluated in $t = 0$ yields the moments of $f_X(x_i)$:

$$\begin{aligned} \left. \frac{d^k}{dt^k} M_X(t) \right|_{t=0} &= \left(\sum_{x_i=X_{min}}^{X_{max}} x_i^k e^{tx_i} f_X(x_i) \right)_{t=0} \\ &= \sum_{x_i=X_{min}}^{X_{max}} x_i^k f_X(x_i) \\ &= E[X^k] \end{aligned} \quad (\text{B.12})$$

Furthermore, based on $M_X(t)$, we can define a second generating function by substituting e^t by z :

$$G_X(z) = \sum_{x_i=X_{min}}^{X_{max}} z^{x_i} f_X(x_i). \quad (\text{B.13})$$

$G_X(z)$ contains the same information as $M_X(t)$. Its derivatives, evaluated in $z = 1$ ($= e^t|_{t=0}$) now yield a set of expected values that are called *factorial moments*:

$$\left. \frac{d^k}{dz^k} G_X(z) \right|_{z=1} = E[X(X-1)\dots(X-k+1)] \quad (\text{B.14})$$

When X can only take positive integer values, $G_X(z)$ is a polynomial of z , with the probabilities $f_X(i) = P(X = i)$ as coefficients. Hence, the derivatives of $G_X(z)$, evaluated in $z = 0$ yield the probabilities:

$$P(X = i) = \frac{1}{i!} \cdot \frac{d^i}{dz^i} G_X(z) \Big|_{z=0}. \quad (\text{B.15})$$

It is this property of $G_X(z)$ that has given it its name of *probability generating function* or *(probability) generating polynomial* of X .

B.3 Functions of discrete random variables

B.3.1 Linear combinations of independent variables

Moment-generating functions are often used to obtain the moments (or the pmf) of linear combinations of independent random variables. Indeed, if X and Y are independent, then the mgf of $U = aX + bY + c$ (with a , b and c constants) is

$$M_U(t) = e^{ct} M_X(t^a) M_Y(t^b). \quad (\text{B.16})$$

When the mgf of X and Y are available as analytical expressions, expression (B.16) yields the moments of $f_U(u)$ in a very elegant way. As a consequence of (B.16), $G_U(z)$ equals

$$G_U(z) = z^c G_X(z^a) G_Y(z^b). \quad (\text{B.17})$$

We will now first consider the case where X and Y are restricted to a finite subset of the integer domain and where a , b and c are also integers. Obviously, as a result, U is integer-valued. In this case, three very simple numerical operations can be combined to obtain the pmf of U . First, the pmf's for $X' = aX$ and $Y' = bY$ must be obtained. We find that:

$$P(X' = i) = \begin{cases} P(X = i/a), & i \bmod a = 0, \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.18})$$

and similarly for Y' . Secondly, we need the pmf of the sum of two independent, integer-valued random variables $V = X' + Y'$. This is given by the *convolution* of the pmf's of X' and Y' :

$$P(V = i) = \sum_{j=X'_{\min}}^{X'_{\max}} P(X' = j) P(Y' = i - j). \quad (\text{B.19})$$

This can be derived in several ways. In [VM00], where X' and Y' were restricted to the positive integer domain, the generating polynomials were elegantly used for this purpose. Indeed, in this case, both $G'_X(z)$ and $G'_Y(z)$ are polynomials in z , and $G_V(z)$ is the *polynomial product* of both. This is equivalent to performing a convolution of both coefficient arrays to obtain the coefficient array (i.e. the pmf) of $G_V(z)$.

Finally, the pmf of $U = V + c$ is given by:

$$P(U = i) = P(V = i - c). \quad (\text{B.20})$$

Note that this is equivalent to performing the convolution B.19 with a pmf that equals one at value c and zero everywhere else.

In general, transformations similar to (B.18), (B.19) and (B.20) can equally be applied when there exists a linear transformation \mathcal{D} of the integer domain

$$\mathcal{D}(i) = \{\alpha i + \beta\}, \quad (\text{B.21})$$

with i integer, α and β real constants and with $X, Y \in \mathcal{D}$ (i.e., the domains of X and Y are subsets of \mathcal{D}). Obviously, in this case, the i in (B.18), (B.19) and (B.20) must be replaced by $\mathcal{D}(i) = \alpha i + \beta$ where necessary, yielding:

$$\left\{ \begin{array}{l} X' = aX \\ \Rightarrow P[X' = \mathcal{D}(i)] = \begin{cases} P[X = \mathcal{D}(i/a)], & i \bmod a = 0, \\ 0, & \text{otherwise} \end{cases} \\ \\ Y' = bY \\ \Rightarrow P[Y' = \mathcal{D}(i)] = \begin{cases} P[Y = \mathcal{D}(i/b)], & i \bmod b = 0, \\ 0, & \text{otherwise} \end{cases} \\ \\ V = X' + Y' \\ \Rightarrow P[V = \mathcal{D}(i)] = \sum_{j=X'_{min}}^{X'_{max}} P[X' = \mathcal{D}(j)] P[Y' = \mathcal{D}(i - j)], \\ \\ U = V + c \\ \Rightarrow P[U = \mathcal{D}(i)] = P[V = \mathcal{D}(i - c)]. \end{array} \right. \quad (\text{B.22})$$

B.3.2 Absolute value of a discrete random variable

The pmf for the discrete random variable $U = |X|$ is given by:

$$P[U = \mathcal{D}(i)] = \begin{cases} P[X = \mathcal{D}(i)] + P[X = -\mathcal{D}(i)], & \mathcal{D}(i) \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.23})$$

B.4 Order statistics

Given a sample of independent observations (X_1, X_2, \dots, X_n) of a discrete random variable X with cumulative distribution function $F_X(x_i)$. These observations can be sorted in ascending order. Now, we can define a new sequence of random variables. Let $X_{(1)}$ be the smallest value, observed across n observations of X , $X_{(2)}$ the next smallest, and so forth. $X_{(n)}$, obviously, is the largest value observed. These new random variables are called the *order statistics* of our sample. Because the observations are independent, the cdf of these order statistics can be computed from the cdf of X . We will restrict our discussion to the *maximum* of the sample $X_{(n)}$, for which the cdf is given by

$$\begin{aligned} F_{X_{(n)}}(x_i) &= P(X_1 \leq x_i; X_2 \leq x_i; \dots; X_n \leq x_i) \\ &= [F_X(x_i)]^n \end{aligned} \quad (\text{B.24})$$

and the *minimum* of the sample $X_{(1)}$, with cdf

$$\begin{aligned} F_{X_{(1)}}(x_i) &= 1 - P(X_1 > x_i; X_2 > x_i; \dots; X_n > x_i) \\ &= 1 - [1 - F_X(x_i)]^n. \end{aligned} \quad (\text{B.25})$$

These expressions can be generalized to the maximum of a set of n independent, but not necessarily equally distributed discrete random variables $(X_I, X_{II}, \dots, X_N)$, where Equation (B.24) becomes the product of the N individual cdf's.

Appendix C

Site distributions and site functions

In the context of interconnect prediction techniques, it is essential to be able to calculate the capacity of the placement architecture for containing wires of different lengths. Mathematically, this capacity is expressed by site functions and site distributions. Their calculation is based on the probabilistic concepts described in Appendix B, but because they are so important, we have dedicated a separate appendix to them.

C.1 Definitions

In a placement grid \mathcal{G} (see Chapter 2), we can define subsets of grid locations or *regions*. Now consider two (not necessarily disjoint) regions \mathcal{G}_1 and \mathcal{G}_2 . For many a priori interconnect prediction techniques, it is important to know the total number of pairs (g_1, g_2) of *distinct* grid locations that are a distance ℓ apart, and for which g_1 is in \mathcal{G}_1 and g_2 in \mathcal{G}_2 .

After placement, such a pair (g_1, g_2) can contain the two gates a two-terminal net is connected to. Hence, it is often referred to as a *(two-terminal) net placement site* or simply a *site*. Hence, the *site function* $\Phi_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ expresses how many different sites of length ℓ are available to connect regions \mathcal{G}_1 and \mathcal{G}_2 .

The normalized version of $\Phi_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ is called the *site distribution* $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$. It can be viewed as the probability mass function (pmf) of the distance between two randomly selected distinct grid locations g_1

and g_2 , belonging to regions \mathcal{G}_1 and \mathcal{G}_2 , respectively.

Obviously, the difference between $\Phi_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ and $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ is merely a normalization constant. However, we will make the distinction between both, depending on how they are calculated and how they are used. When the calculation is done by some sort of enumeration of all sites, we use the site function, while a probabilistic calculation results in the site distribution. Similarly, the site function will typically be used to express numbers of sites, while the site distribution is applied in probabilistic applications.

Finally, note that the suffixes \mathcal{G}_1 and \mathcal{G}_2 will be omitted when there can be no doubt as to which regions are indicated. For instance, when $\mathcal{G}_1 = \mathcal{G}_2 = \mathcal{G}$ (i.e., the entire placement grid), we simply use $\Phi(\ell)$ and $\mathcal{D}(\ell)$.

Obviously, a site function can always be calculated by enumerating all possibilities. However, for a broad range of regions \mathcal{G}_1 and \mathcal{G}_2 , the computational complexity of this enumeration can be considerably reduced by exploiting *symmetry* and by *decomposition*. In particular, for the interconnect prediction techniques presented in this work, \mathcal{G}_1 and \mathcal{G}_2 are *rectangular regions* or else they can be decomposed into a number of such regions. Hence, we will concentrate on the general type of region pairs shown in Figure C.1. Also, we restrict this discussion to two-dimensional placement grids. The extension to three-dimensional grids is straightforward.

A systematic description of enumeration techniques for the derivation of site functions for rectangular and sliceable regions was first given by Van Marck [VM00]. In the next two sections, the probabilistic parallels of his techniques are applied to derive site distributions for two-terminal nets and approximated site distributions for multi-terminal nets.

Also, closed-form expressions for certain site functions were given by Davis et al. (also based on enumeration). A complete derivation of those site functions is given in Section C.4.

C.2 Site distributions for two-terminal nets

Due to the fact that both grid positions g_1 and g_2 are selected with uniform probability, the Manhattan distance ℓ between them can be decomposed into two independent or almost independent components δ_x

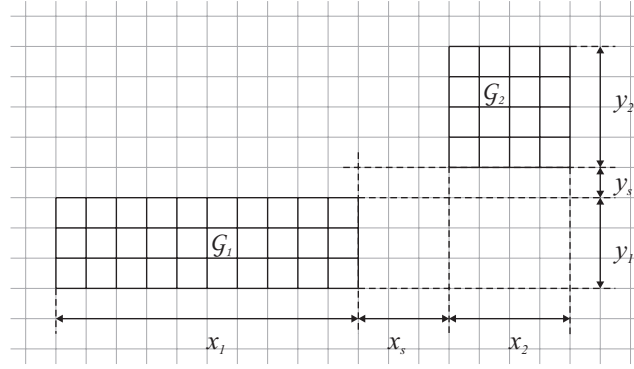


Figure C.1: Graphical illustration of the relevant parameters for calculating site distributions (the case where \mathcal{G}_1 and \mathcal{G}_2 are rectangular regions).

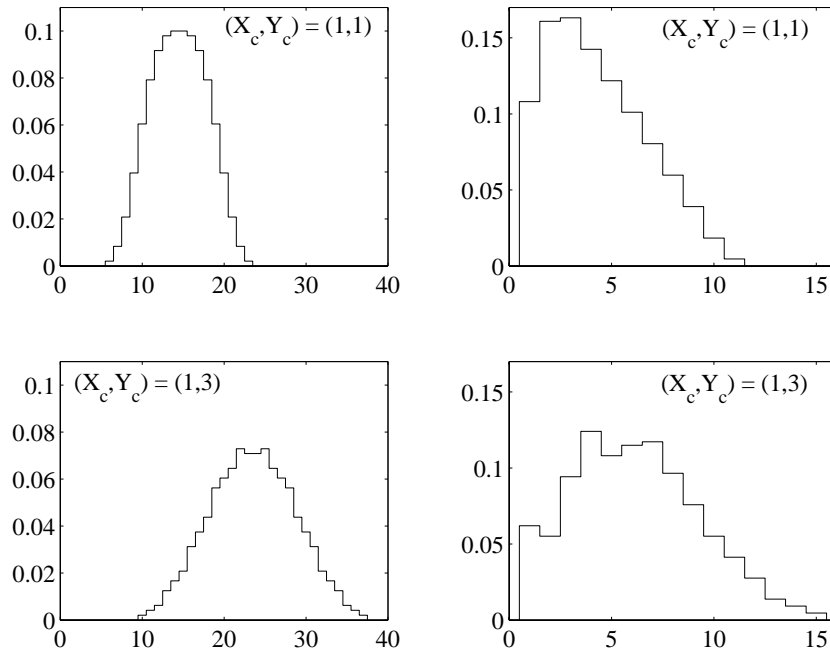


Figure C.2: Some examples of site distributions for the rectangular regions shown in Figure C.1 (In all four plots, the horizontal axis corresponds to the ℓ -values, while their probabilities are shown on the vertical axis): $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_1}(\ell)$ with cell dimensions $(X_c, Y_c) = (1, 1)$ (top left) and $(X_c, Y_c) = (1, 3)$ (bottom left); $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$ with cell dimensions $(X_c, Y_c) = (1, 1)$ (top right) and $(X_c, Y_c) = (1, 3)$ (bottom right).

and δ_y . Indeed, when \mathcal{G}_1 and \mathcal{G}_2 are disjoint, the X and Y -components of the length ℓ are independent. If \mathcal{G}_1 and \mathcal{G}_2 are (partially) overlapping, δ_x and δ_y are constrained by the fact that g_1 and g_2 must be distinct, i.e.:

$$\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(0) = 0 \quad (\text{C.1})$$

This allows us to initially calculate the pmf $P(\delta_x + \delta_y = \ell)$ as if δ_x and δ_y were independent and then correct for the case where $\delta_x = \delta_y = 0$ (i.e., $\ell = 0$):

$$\begin{aligned} \mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(\ell) &= P(\delta_x + \delta_y = \ell \mid \delta_x + \delta_y \neq 0) \\ &= \frac{P[(\delta_x + \delta_y = \ell) \wedge (\delta_x + \delta_y \neq 0)]}{1 - P(\delta_x + \delta_y = 0)}. \end{aligned} \quad (\text{C.2})$$

Numerically, this is equivalent to setting the probability for $\ell = 0$ to zero and renormalizing the resulting pmf.

What remains is to derive the pmf's for δ_x and δ_y . When the placement architecture has non-unit cell dimensions X_c and Y_c , the pmf's for δ_x and δ_y can be calculated first with distances expressed as a number of gate pitches. The transformation to the final length variable, $\ell = X_c \delta_x + Y_c \delta_y$, then requires the application of (B.18) for both length components individually, followed by (B.19) to obtain the sum of both.

Since the derivations for δ_x and δ_y are similar, we will restrict this discussion to δ_x . We will consider δ_x as the length of a (one-dimensional) vector. This vector can be decomposed into three vector components, by using two fixed reference points p_1 and p_2 (the first in \mathcal{G}_1 and the second in \mathcal{G}_2). This is illustrated in Figure C.3. The three components are \vec{D}_1 , \vec{D}_p and \vec{D}_2 . The lengths of both \vec{D}_1 and \vec{D}_2 are integer random variables. Furthermore, p_1 and p_2 have been chosen such that these lengths are uniformly distributed in a well defined domain:

$$\begin{aligned} P(|\vec{D}_1| = d_1) &= \begin{cases} 1/x_1, & d_1 \in [0, x_1 - 1] \\ 0, & \text{otherwise} \end{cases} \\ P(|\vec{D}_2| = d_2) &= \begin{cases} 1/x_2, & d_2 \in [0, x_2 - 1] \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (\text{C.3})$$

Finally, \vec{D}_1 and \vec{D}_2 have the same direction, such that

$$|\vec{D}_1 + \vec{D}_2| = |\vec{D}_1| + |\vec{D}_2|, \quad (\text{C.4})$$

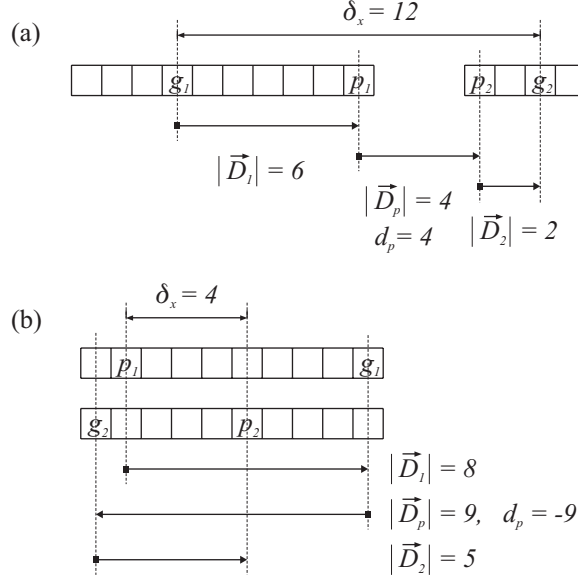


Figure C.3: Two possible situations for decomposition into vector components when calculating the distribution of one-dimensional length components: (a) fully separated x -regions and (b) fully overlapping x -regions.

i.e., the sum of two independent discrete random variables. Finally, let d_p be a constant integer, equal to $|\vec{D}_p|$ if \vec{D}_p has the same direction as \vec{D}_1 and \vec{D}_2 and to $-|\vec{D}_p|$ if it has the opposite direction. The pmf of the random variable δ_x can now be found using the basic operations described in Appendix B:

$$\delta_x = \left| |\vec{D}_1| + |\vec{D}_2| + d_p \right|. \quad (\text{C.5})$$

Note that the absolute value is required, because negative values for d_p would otherwise result in negative lengths.

Putting it all together, we end up with a single numerical recipe for the derivation of $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_2}(\ell)$, *independent* of the sizes and the relative positions of \mathcal{G}_1 and \mathcal{G}_2 and independent of the cell dimensions X_c and Y_c . As an example, we have derived the site distribution $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_1}(\ell)$, for the distance between gate pairs in region \mathcal{G}_1 of Figure C.1, but with $(X_c, Y_c) = (2, 1)$. Figure C.4 summarizes the steps that must be taken to derive the pmf for the X -component δ_x , while Figure C.5 shows how the X and Y -components are combined, as well as the correction to

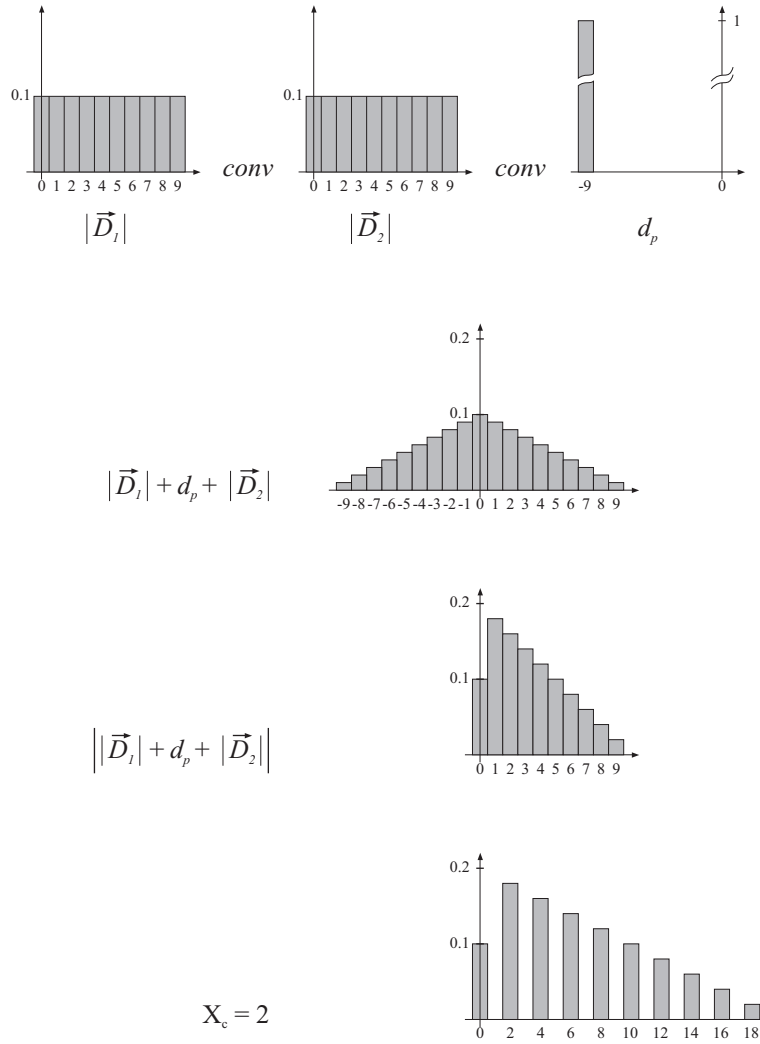


Figure C.4: The different steps involved in calculating the δ_x length distribution for the horizontal length component in $\mathcal{D}_{\mathcal{G}_1, \mathcal{G}_1}(\ell)$, with $(X_c, Y_c) = (2, 1)$. From top to bottom: the pmf's of $|\vec{D}_1|$, $|\vec{D}_2|$ and d_p ; the pmf of their sum (directed); the pmf after taking the absolute value and finally the incorporation of $X_c = 2$.

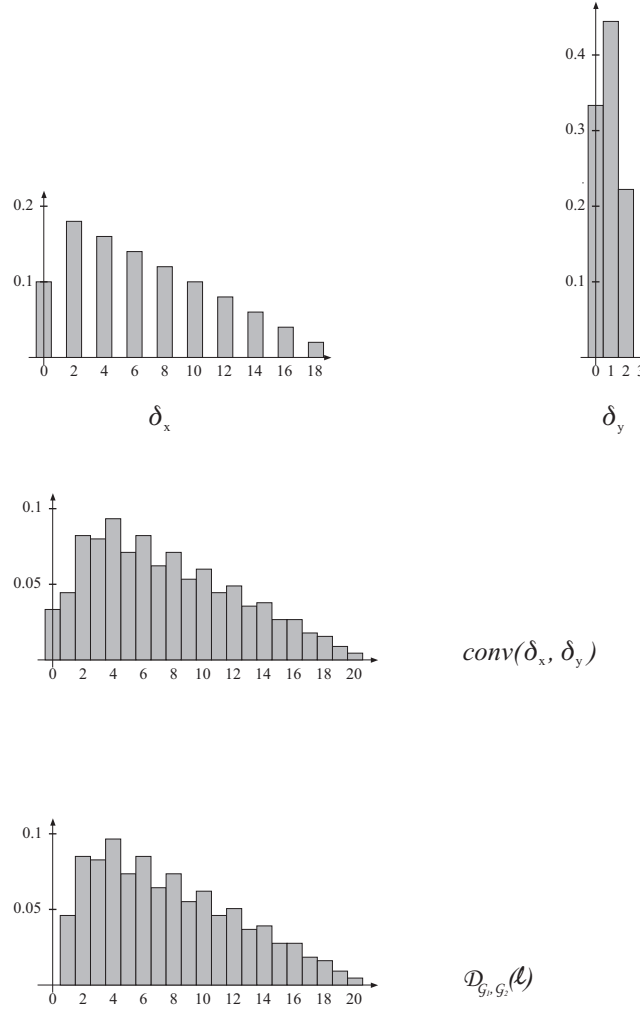


Figure C.5: Combination of the pmf's for the independent δ_x and δ_y length distributions into $\mathcal{D}_{g_1, g_1}(\ell)$, with $(X_c, Y_c) = (2, 1)$. From top to bottom: distributions for each length component; pmf for their sum; pmf after excluding the cases where $\ell = 0$.

exclude $\ell = 0$. Finally, Figure C.2 shows several other site distributions, derived for the rectangles in Figure C.1.

C.3 Site distributions for multi-terminal nets

In this section, we look into the question whether it is possible, with a reasonable amount of computation, to obtain site distributions for k -terminal nets (with $k \geq 2$). However, to do this, we first need to get the definition of such site distributions clear. We could restrict ourselves to the *minimal routing length* of a net as its length metric, given the locations of all gates connected to it. However, as discussed in Chapter 2, this length, also known as the *Steiner minimal routing tree* or SMT-length, cannot be calculated analytically. Because of this, there would seem to be no way of obtaining the pmf for this length in any other way than by (partial) enumeration. Hence, we will extend the definition of site distributions for multi-terminal nets to

the pmf of a well-defined length metric on k -terminal nets, given that the gates connected to those nets are placed randomly at k distinct grid locations.

This definition applies, e.g., to the pmf of HPBB-length (*half perimeter* of the *bounding box*), SSL-length (sum of all *source to sink lengths*) and STT-length (*single trunk tree* length) metrics for randomly placed k -terminal nets (see Chapter 2 for definitions of those length metrics). The single trunk trees considered here are not the optimal ones, but trees for which the *trunk* covers the entire distance in one direction and runs *through* the net source position. The *branches* are orthogonal to the trunk, each connecting it to a single net sink.¹

In the previous section, we derived a general flow for two-terminal nets connecting two rectangular regions. Such a general derivation for multi-terminal nets would be more complicated, since we must ensure that at least one terminal is in each of both regions. The easiest strategy to obtain this would seem to first model the situation where all terminals are homogeneously distributed across both regions, and then exclude the cases where all terminals are in a single region. However, because this would lead us too far, we will restrict this discussion to the

¹Note that it is also possible to approximate the distribution of better (i.e., shorter) STT-variants, but the description of these calculations would lead us too far.

site distributions for the entire placement grid, i.e., $\mathcal{G}_1 = \mathcal{G}_2 = \mathcal{G}$.

To try to obtain multi-terminal net site distributions, essentially the same strategy can be followed as for two-terminal net site distributions. Indeed, for all three length metrics mentioned above, the total length can still be decomposed into separate (one-dimensional) x and y -components. When the pmf's for these components are found, their convolution corresponds to the pmf resulting from *independent* random placement of all gates connected to the net. Then, a correction must be made to exclude all cases where two or more gates are placed on the same site.

C.3.1 One-dimensional length components

In the decomposition of the k -terminal net site distributions for HPBB, SSL and STT-lengths, only two different kinds of one-dimensional length components occur. The first (henceforth called a *type I* length component) is the sum of all (one-dimensional) source-to-sink length components in the net, while the second (a *type II* length component) is the maximal distance between any two gates connected to it. In the HPBB-metric, both length components are of type II while in the SSL-metric they are both of type I. For the STT-metric, the trunk length is of type II, while the total length of the branches is of type I. Hence, two types of pmf's can be defined for STT's: the pmf for x -trunk trees and the pmf for y -trunk trees. The minimum of both can not be found probabilistically, since it depends on the specific gate locations for each individual net placement, but it is possible to select the pmf with the smallest mean.

For all three length metrics, the pmf can first be calculated assuming a unit pitch between neighboring gate locations. Then, before convolving the pmf's for each dimension, the real X_c and Y_c must be taken into account. Also, as we did for two-terminal nets, we only discuss both types of length components for the x -dimension.

Type I length components

We initially assume that the x -positions x_1, x_2, \dots, x_k of the k gates connected to the net are independent and uniformly distributed integer random variables in the range $1 \dots X_g$.

For the SSL-component, we need an expression for the pmf of the

sum of the $k - 1$ independent random variables, expressing the distances between the source and the $k - 1$ sinks. This pmf must be obtained for all possible source positions, which are equally likely, with probability $1/X_g$. Hence, the pmf for each source position is found by performing a $(k - 1)$ -way convolution, expressing the sum of the distances of the $k - 1$ sinks to the source. Then, these partial pmf's are multiplied by $1/X_g$ and added.

Type II length components

The HPBB-component is a bit more tricky. We need to obtain the pmf of the random variable $\delta_x = x_{max} - x_{min}$ with $x_{max} = \max(x_1, x_2, \dots, x_k)$ and $x_{min} = \min(x_1, x_2, \dots, x_k)$. The easiest way to compute this pmf, is through its cumulative distribution $P(\delta_x \leq i)$. Indeed, this is given by the probability that *all* k vertices are located in a window of size $i + 1$, multiplied by the number of positions for such a window. Because of the uniform distribution of the block positions, the probability that the x -coordinate of any given vertex is within a specific window of size $i + 1$ equals

$$\frac{i + 1}{X_g}.$$

Since the vertex positions are assumed independent, the probability that all vertices are within that same window is

$$\left(\frac{i + 1}{X_g} \right)^k.$$

After multiplication by the number of window positions $(X_g - i)$, we find:

$$P(\delta_x \leq i) = (X_g - i) \left(\frac{i + 1}{X_g} \right)^k, \quad (\text{C.6})$$

from which the pmf can be obtained.

C.3.2 Coinciding gate locations

As it turns out, this correction is relatively easy to perform for the HPBB-metric, but extremely complex for both the SSL and the STT metrics. However, we have found that without this correction, the deviations are very small. This can be understood from the following arguments:

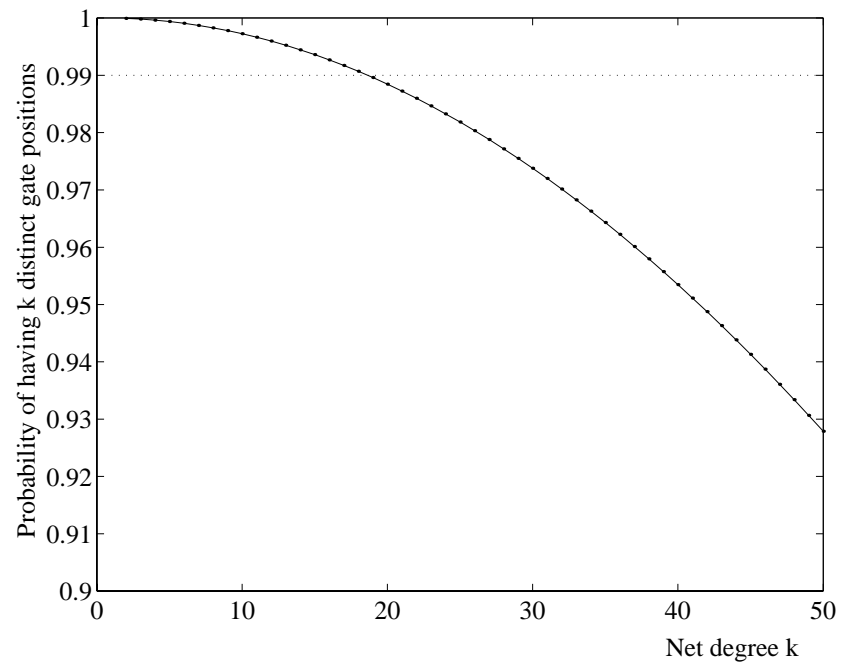


Figure C.6: Probability of k gates occupying distinct grid positions when placed independently in a grid of 128×128 .

- In grids of realistic size, the probability of two or more gates coinciding is relatively small. Figure C.6 shows that, for a placement grid of 128×128 , this probability remains below 1% for up to 18-terminal nets. For larger placement grids, this probability rapidly decreases.
- For type I length components, there is no reason to assume that the pmf for the cases where one or more gate positions coincide is drastically different from the pmf for distinct gate positions. Hence, if we do not exclude those cases, they will not have a large impact on the site distribution.
- For type II length distributions, the number of distinct gate positions does matter, since it directly affects the maximum spanning distance. However, while this maximum increases rapidly for small number of gate positions, it tends to saturate as the number of gate positions grows large.

Hence, calculating site distributions with independent length components can be considered a good approximation of the real site distributions. Figure C.7 illustrates this for 5- and 10-terminal nets. We have measured the HPBB, SMT and SSL site distributions for a placement grid of 128×128 by performing 100000 random placements of a single 5-terminal net (10-terminal net). These are compared to the approximated site distributions for the HPBB, SSL and STT length metrics.² Figure C.8 shows the measured and estimated means of the site distributions for the same length metrics, for net degrees ranging from 2 to 20. In this figure, the deviation caused by our approximation is barely visible. We can conclude from these Figures that there is indeed a very large difference between the SMT, SSL, HPBB and STT length metrics for multi-terminal nets. It appears that the HPBB-metric is closest to the Steiner minimal tree length.

²Due to lack of time, STT was not measured. However, because it consists of the combination of a type I and a type II length component, we can assume that for this metric, the quality of our estimations is comparable to that for the SSL and HPBB metrics.

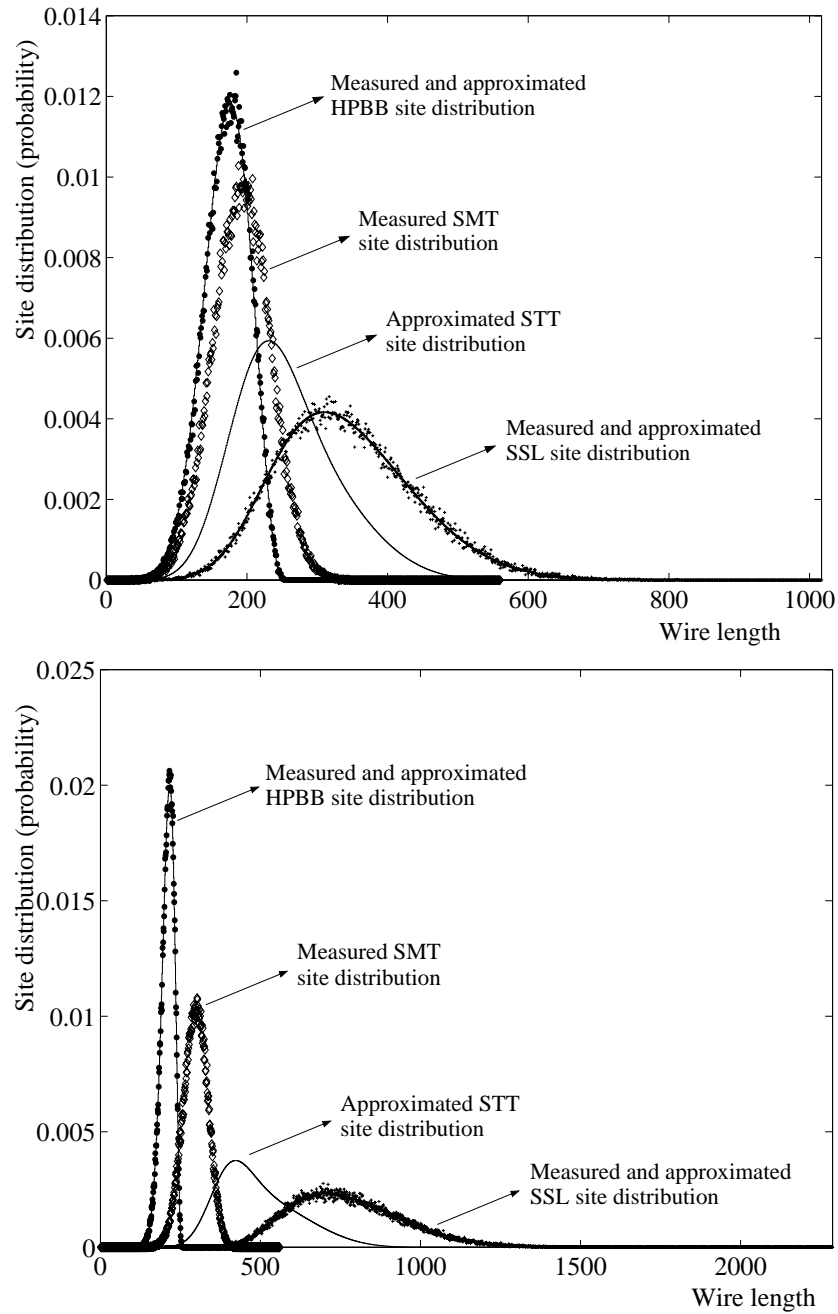


Figure C.7: Measured and approximated site distributions for different length metrics of 5-terminal nets in a 128×128 placement grid (top) and 10-terminal nets (bottom).

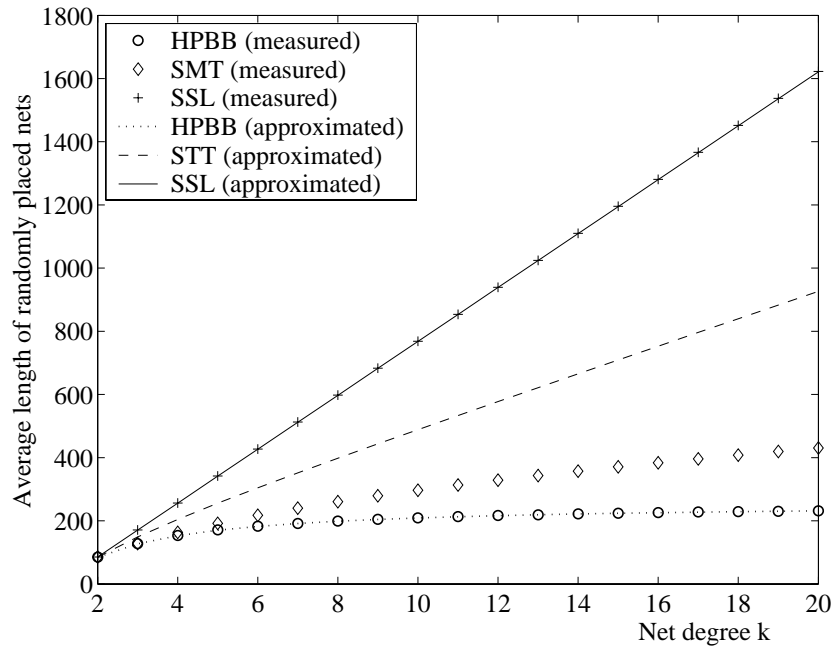


Figure C.8: Measured and approximated average wire lengths for placement in a 128×128 placement grid, as a function of the net degree k .

C.3.3 Approximation of multi-terminal wire length distributions

Approximation of the multi-terminal SSL distribution

Assume that we know the distribution of source-to-sink lengths for a given circuit, resulting from a placement process \mathcal{P} that minimizes the total source-to-sink wire length. Under the homogeneity assumption, it is possible to estimate the distribution of the SSL length metric for k -terminal nets for which k is relatively small, compared to the size of the placement grid. Indeed, in this case, the positions of the $k - 1$ gates to which the net sinks are connected are almost independent random variables. Hence, the distribution of the total SSL-length of a k -terminal net can be approximated by the $(k - 1)$ -fold convolution of the source-to-sink length distribution.

Conversion of SLL to other length metrics

If we can derive an average relation between SSL and other multi-terminal net length metrics, such as SMT, HPBB or STT, as a function of the net degree k :

$$\begin{aligned}\ell_{SMT}(k) &= f_{SMT}(\ell_{SSL}, k) \\ \ell_{HPBB}(k) &= f_{HPBB}(\ell_{SSL}, k) \\ \ell_{STT}(k) &= f_{STT}(\ell_{SSL}, k)\end{aligned}$$

it can be applied to convert the SSL-distribution to the length distribution under one of these other length metrics.

The approximated multi-terminal net site distributions can be used to derive the transformations $f_{metric}(\ell_{SSL}, k)$ based on their cumulative distributions by requiring that for every wire length ℓ_{SSL} :

$$\sum_{i=1}^{f_{metric}(\ell_{SSL}, k)} \mathcal{D}_{metric}(i, k) = \sum_{i=1}^{\ell_{SSL}} \mathcal{D}_{SSL}(i, k) \quad (C.7)$$

The derivation of this transformation is illustrated in Figure C.9 for the HPBB length metric.

From the circuit topology, the number of nets of each net degree can easily be extracted. The combination of this information with transformation (C.7) for each net degree leads to an approximated wire length

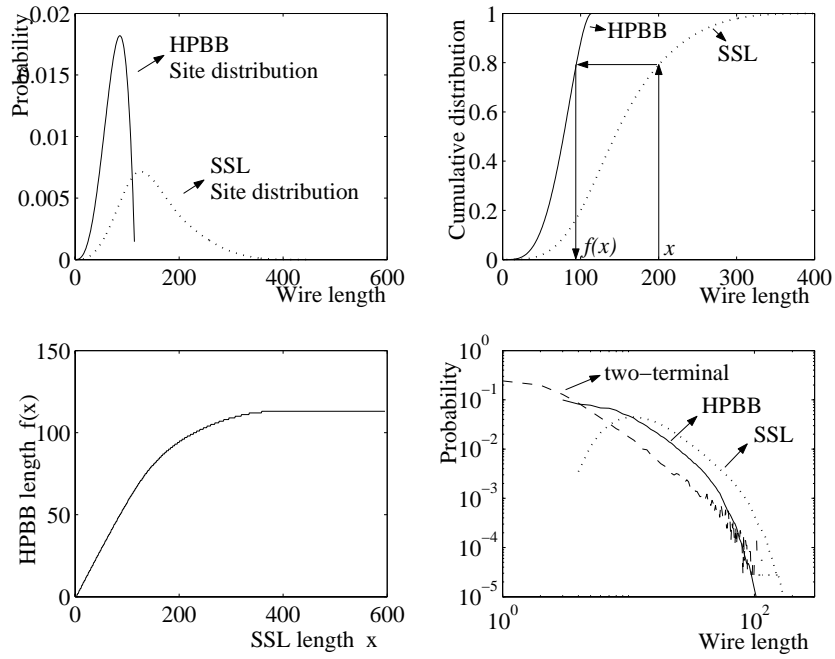


Figure C.9: Derivation of the HPBB length distribution for 5-terminal nets (benchmark `ibm01`, square grid/square cell placement with SA): (top left) approximated SSL and HPBB site distributions for 5-terminal nets; (top right) cumulative distributions and derivation of the transformation; (bottom left) resulting transformation $f_{HPBB}(\ell_{SSL}, 5)$; (bottom right) application of this transformation to approximate the HPBB length distribution of 5-terminal nets (SA-placement of benchmark `ibm01`).

distribution for circuits with multi-terminal nets (under the homogeneity assumption). Remember that in the derivation of this approximated distribution, we also assumed that the total SSL-length was optimized during placement. To validate this approximated distribution, we have performed two SA placements of the original version of benchmark `ibm01` (i.e., *with* multi-terminal nets)³, one where the SSL-metric was optimized and one with the total HPBB-length optimized. For each of these placements, we have measured the length distribution of all source-to-sink pairs and used it for estimating the HPBB-length distribution. The SMT, HPBB and SSL length distributions were also measured from both placements. Figures C.10 and C.11 show the resulting wire length distributions, as well as the average wire length of the nets as a function of the net degree.

From these figures, we can conclude that the approximation is relatively good when SSL is optimized during placement. However, it is very far from the measured results for the placement where HPBB was optimized. This is due to the nature of HPBB optimization: as long as the bounding box does not change, the position of the terminals within the bounding box does not matter. Hence, they can be assumed to be random. As a result, the homogeneity assumption is no longer fulfilled, since the source-to-sink lengths that do not determine the net's bounding box are not further optimized because they are not *visible* in the optimization cost function. As the net degree increases, relatively fewer source-to-sink pairs are visible in the cost function, so the average wire length for large-fanout nets will be larger than that for small-fanout nets. Because our estimation applies the total source-to-sink length distribution for all net degrees, it results in a large overestimate of the lengths of small-fanout nets.

In state-of the art placement tools, either HPBB or a metric that is even closer to SMT is minimized. The example above shows that, at least for predicting multi-terminal wire lengths, the homogeneity assumption for source-to-sink lengths is no longer valid.

³To save computation time, the experiments for this section were performed with a much higher cooling rate than the rest of the SA experiments (0.98 instead of 0.999)

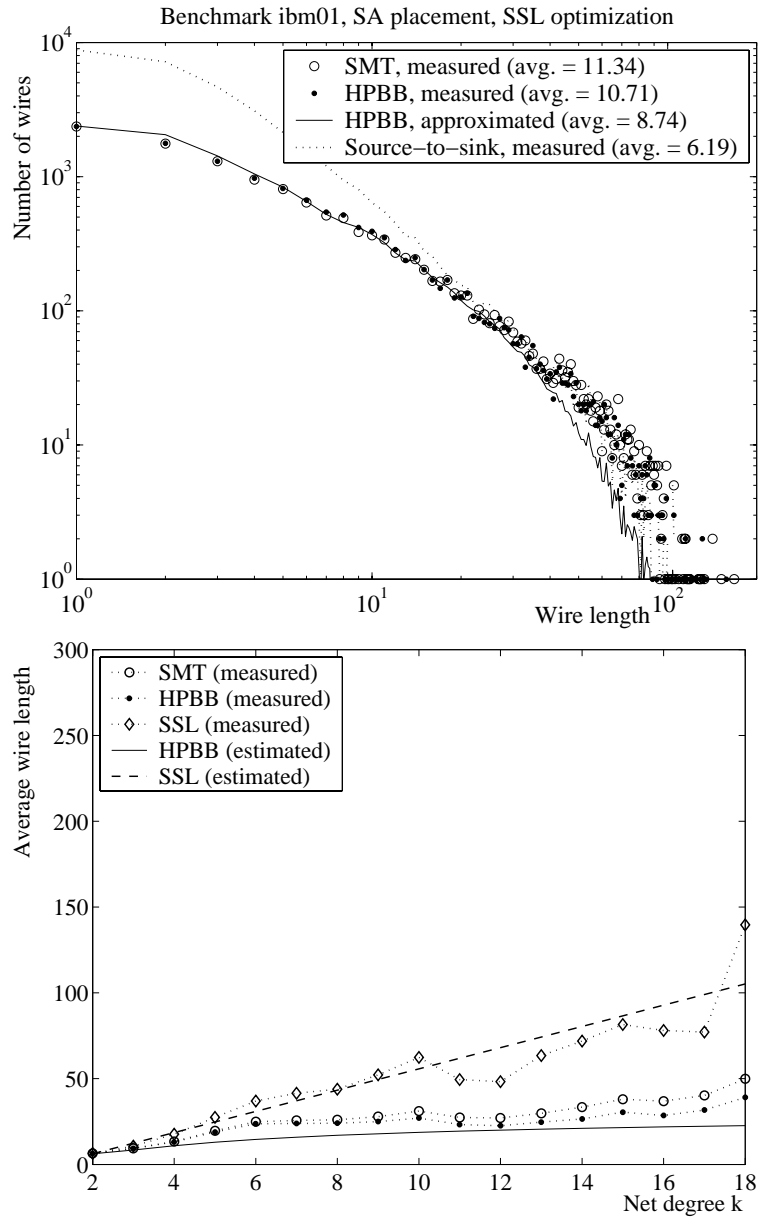


Figure C.10: Results from an SA placement of the multi-terminal net version of benchmark *ibm01*, with total SSL optimized during placement: (top) measured SMT, HPBB and source-to-sink length distributions and approximated HPBB length distribution; (bottom) measured (SSL, HPBB, SMT) and approximated (SSL, HPBB) average wire length as a function of net degree.

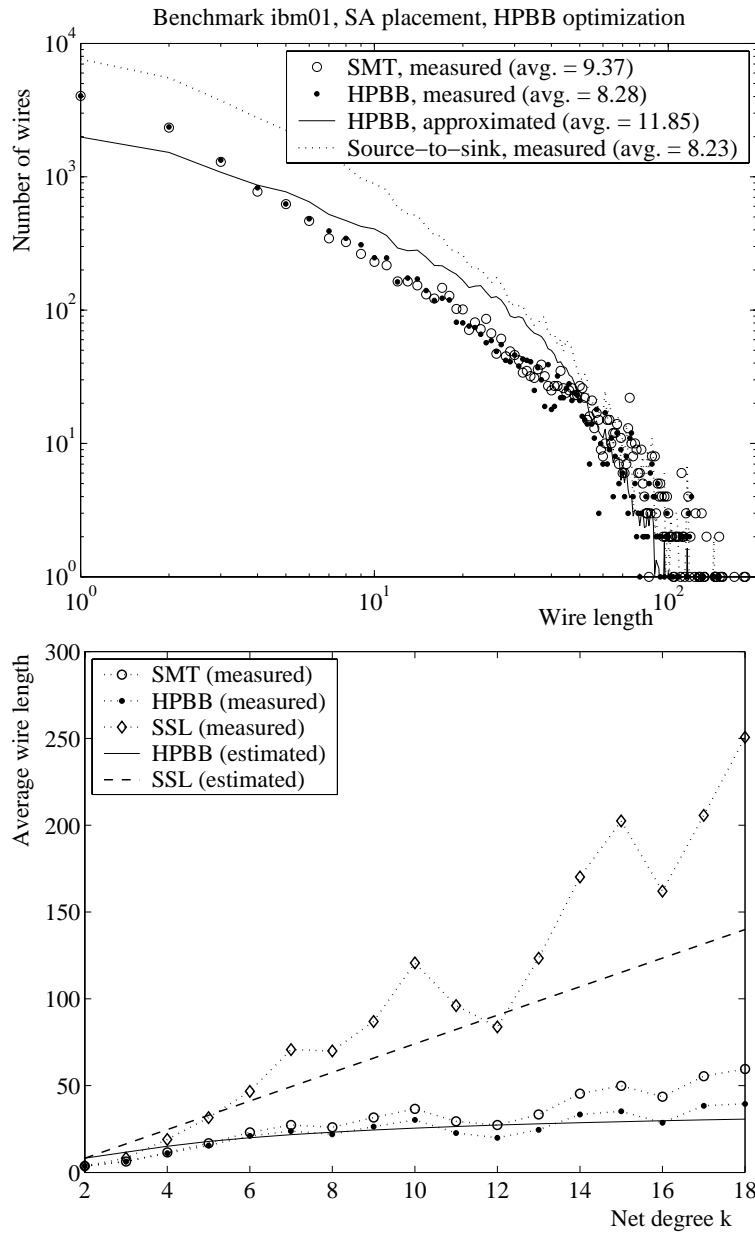


Figure C.11: Results from an SA placement of the multi-terminal net version of benchmark *ibm01*, with total HPBB length optimized during placement: (top) measured SMT, HPBB and source-to-sink length distributions and approximated HPBB length distribution; (bottom) measured (SSL, HPBB, SMT) and approximated (SSL, HPBB) average wire length as a function of net degree.

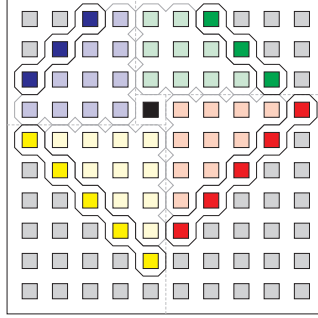


Figure C.12: Division of the layout grid into four quadrants around coordinates (j, i) to derive the local site function (illustrated for $\ell = 5$).

	rows	columns
First quadrant (I, green)	$1 \cdots i - 1$	$j \cdots Y_g$
Second quadrant (II, blue)	$1 \cdots i$	$1 \cdots j - 1$
Third quadrant (III, yellow)	$i + 1 \cdots Y_g$	$1 \cdots j$
Fourth quadrant (IV, red)	$i \cdots Y_g$	$j + 1 \cdots X_g$

Table C.1: Division of the layout grid into quadrants: row and column ranges for each quadrant. Colors correspond to the colors used for each quadrant in Figure C.12.

C.4 Local site functions in a rectangular grid

In Davis' technique for predicting wire length distributions, local site functions $\Phi(i, j, \ell)$ are used. These functions express the number of grid locations at a distance ℓ from the grid location in row i and column j (i.e., at coordinates $(x, y) = (j, i)$). In his original paper [DDM98], Davis gave analytical expressions for these local site functions. Unfortunately, some confusion seems to have arisen regarding their correct form (cf. [Chr01]). Therefore, a systematic derivation of $\Phi(i, j, \ell)$ is given below. Our approach is based on dividing the layout grid into quadrants around coordinates (i, j) as is shown in figure C.12. Using the goniometric conventions for numbering the quadrants, the ranges for each quadrant are indicated in Table C.1. We derive an analytical expression $\Phi_{XX}(i, j, \ell)$ ($XX \in \{I, II, III, IV\}$) for the number of gate locations in each quadrant, at a distance ℓ from a given location (j, i) . The function $\Phi(i, j, \ell)$, as it was defined by Davis, is then given by $\Phi_{III}(i, j, \ell) + \Phi_{IV}(i, j, \ell)$.

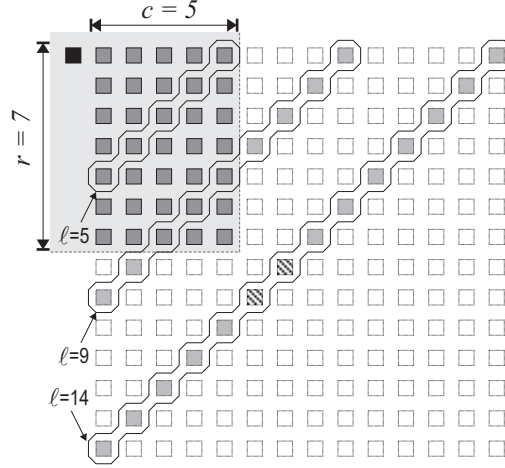


Figure C.13: Graphical illustration of some of the possible cases for the relation between ℓ , r and c in a single quadrant. For $\ell = 14$, the dashed gate locations are first subtracted twice and then added once to obtain the correct final value of zero.

Without loss of generality, we derive $\Phi_{III}(i, j, \ell)$. To ease our notation for now, we refer to the number of rows in the quadrant as r ($r \in [1, Y_g]$) and to the number of columns as c ($c \in [1, X_g]$). The quadrant under consideration can then be seen as a rectangular corner region of size $r \times c$ in a much larger grid. This is illustrated in Figure C.13. In this larger grid, the number of locations at distance ℓ from the black gate location equals ℓ . From this number, we need to subtract the gate locations at distance ℓ that fall outside the corner region. Depending on the relationship between r , c and ℓ , there are several possibilities:

$$\Phi_{quad} = \begin{cases} \ell, & \ell \leq \min(r, c) \\ \ell - (\ell - r), & r < \ell < c \\ \ell - (\ell - c), & c < \ell < r \\ \ell - (\ell - r) - (\ell - c), & \max(r, c) < \ell < r + c \\ \ell - (\ell - r) - (\ell - c) + (\ell - r - c), & \ell > r + c \end{cases} \quad (\text{C.8})$$

In the last case, some gates have been subtracted twice and need to be added again. If we introduce the step function $\mu_0(XX)$, which is one if its argument is strictly positive and zero otherwise, all cases in Equation C.8 can be combined into a single analytical expression:

$$\Phi_{quad} = \ell \mu_0(\ell) - (\ell - r) \mu_0(\ell - r) - (\ell - c) \mu_0(\ell - c)$$

$$+(\ell-r-c)\mu_0(\ell-r-c) \quad (\text{C.9})$$

The resulting equation is symmetrical in r and c . Hence, the Φ -functions for all four quadrants can be obtained from it, by simply substituting the proper values for r and c (these can be derived from Table C.1). We find:

$$\begin{aligned} \Phi_I(i, j, \ell) = & \ell\mu_0(\ell) - (\ell-i+1)\mu_0(\ell-i+1) \\ & - (\ell-X_g+j-1)\mu_0(\ell-X_g+j-1) \\ & + (\ell-i+1-X_g+j-1)\mu_0(\ell-i+1-X_g+j-1) \end{aligned} \quad (\text{C.10})$$

$$\begin{aligned} \Phi_{II}(i, j, \ell) = & \ell\mu_0(\ell) - (\ell-i)\mu_0(\ell-i) \\ & - (\ell-j+1)\mu_0(\ell-j+1) \\ & + (\ell-i-j+1)\mu_0(\ell-i-j+1) \end{aligned} \quad (\text{C.11})$$

$$\begin{aligned} \Phi_{III}(i, j, \ell) = & \ell\mu_0(\ell) - (\ell-Y_g+i)\mu_0(\ell-Y_g+i) \\ & - (\ell-j)\mu_0(\ell-j) \\ & + (\ell-Y_g+i-j)\mu_0(\ell-Y_g+i-j) \end{aligned} \quad (\text{C.12})$$

$$\begin{aligned} \Phi_{IV}(i, j, \ell) = & \ell\mu_0(\ell) - (\ell-Y_g+i-1)\mu_0(\ell-Y_g+i-1) \\ & - (\ell-X_g+j)\mu_0(\ell-X_g+j) \\ & + (\ell-Y_g+i-1-X_g+j)\mu_0(\ell-Y_g+i-1-X_g+j) \end{aligned} \quad (\text{C.13})$$

The Φ -function required for Davis' technique is given by:

$$\begin{aligned} \Phi(i, j, \ell) = & \Phi_{III}(i, j, \ell) + \Phi_{IV}(i, j, \ell) \\ = & 2\ell\mu_0(\ell) \\ & - (\ell-Y_g+i)\mu_0(\ell-Y_g+i) \\ & - (\ell-j)\mu_0(\ell-j) \\ & + (\ell-Y_g+i-j)\mu_0(\ell-Y_g+i-j) \\ & - (\ell-Y_g+i-1)\mu_0(\ell-Y_g+i-1) \\ & - (\ell-X_g+j)\mu_0(\ell-X_g+j) \\ & + (\ell-Y_g+i-1-X_g+j)\mu_0(\ell-Y_g+i-1-X_g+j) \end{aligned} \quad (\text{C.14})$$

Note that, in contrast to the expressions in either [DDM98] or [Chr01], our Φ -functions were derived in such a way that they are directly applicable to non-square grids. However, our closed-form expression for $\Phi(i, j, \ell)$ applies only to placement grids with $X_c = Y_c = 1$. Local site functions for placement grids with non-square cells can be calculated through the site distributions, using the the numerical probabilistic techniques described in Section C.2.

Bibliography

- [AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
- [ABZ⁺02] Aseem Agarwal, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Min Zhao, Kaushik Gala, and Rajendran Panda. Path-based statistical timing analysis considering inter- and intra-die correlations. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 16–21. ACM/SIGDA, 2002.
- [ABZV02] Aseem Agarwal, David Blaauw, Vladimir Zolotov, and Sarma Vrudhula. Statistical timing analysis using bounds and selective enumeration. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 29–36. ACM/SIGDA, 2002.
- [ADN92] Pranav Ashar, Srinivas Devadas, and Richard A. Newton. *Sequential Logic Synthesis*. Kluwer Academic Publishers, 1992.
- [ADQ99a] C. J. Alpert, A. Devgan, and S. T. Quay. Buffer insertion for noise and delay optimization. *IEEE Transactions on Computer-Aided Design*, 18(11):1633–1645, 1999.
- [ADQ99b] C. J. Alpert, A. Devgan, and S. T. Quay. Buffer insertion with accurate gate and interconnect delay computation. In *Proceedings of the 39th ACM/IEEE Design Automation Conference (DAC)*, pages 479–484, 1999.

- [Alp98] C. J. Alpert. The ISPD98 circuit benchmark suite. In *Proc. of the 1998 Intl. Symp. on Physical Design*, pages 80–85. ACM Press, April 1998. Available at <http://vlsicad.cs.ucla.edu/~cheese/ispd98.html>.
- [AN03] J.H. Anderson and F.N. Najm. Switching activity analysis and pre-layout activity prediction for fpgas. In *Proceedings of the International Workshop on System-Level Interconnect Prediction*, pages 15–21. ACM/SIGDA, 2003.
- [AYNM02] Ei Ando, Masafumi Yamashita, Toshio Nakata, and Yusuke Matsunaga. The statistical longest path problem and its application to delay analysis of logical circuits. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 134–139. ACM/SIGDA, 2002.
- [Ber97] Michel Berkelaar. Statistical delay calculation, a linear time method. In *Proceedings of TAU 97*, 1997.
- [BN01] Srinivas Bodapati and Farid M. Najm. Prelayout estimation of individual wire lengths. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(6):943–958, 2001.
- [Boe01] Stefan Boettcher. Extremal optimization for graph partitioning. *Physical Review E*, 64, 2001.
- [Bol85] B. Bollobás. *Random graphs*. Academic Press, London, 1985.
- [BR97] Vaughn Betz and Jonathan Rose. VPR: A new packing, placement and routing tool for FPGA research. In *Proceedings of the Seventh International Workshop on Field-Programmable Logic and Applications*, pages 213–222, 1997.
- [Cat00] Francky Catthoor, editor. *Unified low-power design flow for data-dominated multi-media and telecom applications*. Kluwer Academic Publishers, 2000.
- [CBL] NCSU Collaborative Benchmarking Laboratory (CBL). Available at <http://www.cbl.ncsu.edu/>.

- [CC91] J. E. Cotter and P. Christie. The analytical form of the length distribution function of computer interconnections. *IEEE Trans. Circuits & Syst.*, 38:317–320, 1991.
- [CCK⁺00] A.E. Caldwell, Y. Cao, A.B. Kahng, F. Koushanfar, H. Lu, I.L. Markov, M. Oliver, D. Stroobandt, and D. Sylvester. GTX: The MARCO GSRC technology extrapolation system. In *IEEE/ACM Design Automation Conf.*, June 2000. Web: <http://vlsicad.cs.ucla.edu/GSRC/GTX/>.
- [CCMS00] Olivier Coudert, Jason Cong, Sharad Malik, and Majid Sarrafzadeh. Incremental CAD. In *Proceedings of the 2000 International Conference on CAD*, pages 236–243, 2000.
- [CHA⁺92] Jue-Hsien Chern, Jean Huang, Lawrence Arledge, Ping-Chung Li, and Ping Yang. Multilevel metal capacitance models for CAD design synthesis systems. *IEEE Electronic Device Letters*, 13(1):32–34, 1992.
- [Chr00] P. Christie. Rent exponent prediction methods. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 8(6):679–686, December 2000.
- [Chr01] P. Christie. A differential equation for placement analysis. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 9(6):913–921, December 2001.
- [CKL01] Chung-Kuan Cheng, Andrew B. Kahng, and Bao Liu. Interconnect implications of growth-based structural models for VLSI circuits. In *Proceedings of the Workshop on System Level Interconnect Prediction (SLIP)*, pages 99–106. ACM/SIGDA, ACM Press, 2001.
- [CKLS02] Chung-Kuan Cheng, Andrew B. Kahng, Bao Liu, and Dirk Stroobandt. Towards better wireload models in the presence of obstacles. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 10(2):177–189, 2002.
- [CL00] Jason Cong and Sung Kyu Lim. Physical planning with retiming. In *Proceedings of the 2000 International Conference on CAD*, pages 2–7, 2000.

- [Con01] Jason Cong. An interconnect design flow for nanometer technologies. *Proceedings of the IEEE*, 89(4):505–525, 2001.
- [Cou96] Olivier Coudert. Gate sizing: a general purpose optimization approach. In *Proceedings of the 1996 Design Automation Conference (DAC)*, pages 734–739, 1996.
- [CP99] Jason Cong and David Zhigang Pan. Interconnect delay estimation models for synthesis and design planning. In *Proceedings of the Asia and South Pacific Design Automation Conference*, pages 97–100, 1999.
- [CP01] Phillip Christie and José Pineda de Gyvez. Pre-layout prediction of interconnect manufacturability. In *Proceedings of the 2001 International Workshop on System Level Interconnect Prediction*, pages 167–173. ACM/SIGDA, ACM Press, 2001.
- [CPO02] Mustafa Celik, Lawrence Pileggi, and Altan Odabasioglu. *IC Interconnect Analysis*. Kluwer Academic Publishers, 2002.
- [CQZC02] Hongyu Chen, Changge Qiao, Feng Zhou, and Chung-Kuan Cheng. Refined single trunk tree: A rectilinear steiner tree generator for interconnect prediction. In *Proceedings of the Workshop on System Level Interconnect Prediction (SLIP)*, pages 85–89. ACM/SIGDA, ACM Press, 2002.
- [CS00] P. Christie and D. Stroobandt. The interpretation and application of Rent’s rule. *IEEE Trans. on VLSI Systems, Special Issue on System-Level Interconnect Prediction*, 8(6):639–648, December 2000.
- [CXS00] Chunhong Chen, Yang Xiaojian, and Majid Sarrafzadeh. Potential slack, an effective metric of combinational circuit performance. In *Proceedings of the International Conference on Computer Aided Design (ICCAD)*, pages 198–201, 2000.
- [DBMVC01] J. Dambre, M. Brunfaut, W. Meeus, and J. Van Campenhout. Impact of optical I/O on FPGA electronic routing

- delays. In *Proceedings of the 27th European Conference on Optical Communication (ECOC'01)*, volume 3, pages 294–295. ACM/SIGDA, 2001.
- [DDM98] J. A. Davis, V. K. De, and J. D. Meindl. A stochastic wire-length distribution for gigascale integration (GSI) – PART I: Derivation and validation. *IEEE Trans. on Electron Devices*, 45(3):580–589, March 1998.
- [Don79] W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Trans. Circuits & Syst.*, CAS-26:272–277, 1979.
- [Don81] W. E. Donath. Wire length distribution for placements of computer logic. *IBM J. of Research and Development*, 25:152–155, 1981.
- [DSVC02] J. Dambre, D. Stroobandt, and J. Van Campenhout. A probabilistic approach to clock cycle prediction. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 9–15. ACM/SIGDA, 2002.
- [DSVC03a] J. Dambre, D. Stroobandt, and J. Van Campenhout. Fast estimation of the partitioning Rent characteristic, using a recursive partitioning model. In *Proceedings of the International Workshop on System-Level Interconnect Prediction*, pages 45–52. ACM/SIGDA, 2003.
- [DSVC03b] J. Dambre, D. Stroobandt, and J. Van Campenhout. Improvements to Donath’s hierarchical model for interconnect prediction in VLSI circuits. *submitted to IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2003.
- [DVSVC01] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. On Rent’s rule for rectangular regions. In *Proc. of Intl. Workshop on System-Level Interconnect Prediction*, pages 49–56, March 2001.
- [DVSVC02] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. Getting more out of Donath’s hierarchical model for interconnect prediction. In *Proc.*

- of Intl. Workshop on System-Level Interconnect Prediction*, pages 9–16, April 2002.
- [DVSVC03] J. Dambre, P. Verplaetse, D. Stroobandt, and J. Van Campenhout. A comparison of various terminal-gate relationships for interconnect prediction in VLSI circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Special issue on System-Level Interconnect Prediction*, 11(1):24–34, 2003.
- [DVVC99a] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the impact of optical interconnect latency on the performance of optoelectronic FPGAs. In *Workshop notes of the 3rd IEEE Workshop on Signal Propagation on Interconnects*. IEEE, 1999.
- [DVVC99b] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the impact of optical interconnect latency on the performance of optoelectronic FPGAs. In *Proceedings of the Sixth International Conference on Parallel Interconnects*, pages 91–97. IEEE Computer Society, 1999.
- [DVVC00] J. Dambre, H. Van Marck, and J. Van Campenhout. Quantifying the performance of optoelectronic FPGAs: the impact of optical interconnect latency. *Interconnects in VLSI Design*, pages 195–202, 2000.
- [ER60] P. Erdős and A. Rényi. On the evolution of random graphs. *Mat. Kutató Int. Közl*, 5:17–60, 1960.
- [Erd59] P. Erdős. Graph theory and probability. *Canad. Math*, 11:34–38, 1959.
- [FA86] Yaotian Fu and P.W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimization. *Journal of Physics A: Math. Gen.*, 19:1605–1620, 1986.
- [Feu82] Michael Feuer. Connectivity of random logic. *IEEE Trans. on Computers*, c-31(1):29–33, 1982.
- [GA89] Carol V. Gura and Jacob A. Abraham. Average interconnection length and interconnection distribution based

- on Rent's rule. In *Proceedings of DAC*, pages 574–577, 1989.
- [GK83] D. Gajski and R. Kuhn. New VLSI tools, guest editor's introduction. *IEEE Computer*, 6(12):11–14, 1983.
- [GKSV01] Wilsin Gosti, Sunil Khatri, and Alberto Sangiovanni-Vincentelli. Addressing the timing closure problem by integrating logic optimization and placement. In *Proceedings of the 2001 International Conference on CAD*, pages 224–230, 2001.
- [GNBSV98] W. Gosti, A. Narayan, R. K. Brayton, and A. Sangiovanni-Vincentelli. Wireplanning in logic synthesis. In *Proceedings of the 1998 International Conference on CAD*, pages 26–33, 1998.
- [HKKR94] L. Hagen, A. B. Kahng, F. J. Kurdahi, and C. Ramachandran. On the intrinsic Rent parameter and spectrabased partitioning methodologies. *IEEE Trans. on Comput.-Aided Des., Integrated Circuits & Syst.*, 13(1):27–37, January 1994.
- [HO00] Masanori Hashimoto and Hidetoshi Onodera. A performance optimization method by gate sizing using statistical timing analysis. In *Proceedings of the International Symposium on Physical Design*, pages 111–116. ACM Press, 2000.
- [HS96] Gary D. Hachtel and Fabio Somenzi. *Logic Synthesis and Verification Algorithms*. Kluwer Academic Publishers, 1996.
- [IK01] Neumann Ingmar and Wolfgang Kunz. Placement driven retiming with a coupled edge timing model. In *Proceedings of the 2001 International Conference on CAD*, pages 95–102, 2001.
- [ISHC02] Muzammil Iqbal, Ahmed Sharkawy, Usman Hameed, and Phillip Christie. Stochastic wire length sampling for cycle time estimation. In *Proceedings of the 2002 International Workshop on System Level Interconnect Prediction*, pages 91–96. ACM/SIGDA, ACM Press, 2002.

- [ITR01] Semiconductor industry association (SIA): International technology roadmap for semiconductors, December 2001. Available at <http://public.itrs.net/>.
- [JB00] E.T.A.F. Jacobs and M.R.C.M. Berkelaar. Gate sizing using a statistical delay model. In *Proceedings of the Design Automation and Test in Europe Conference*, pages 283–290, 2000.
- [KK98] G. Karypis and V. Kumar. *hMetis: A Hypergraph Partitioning Package*, November 1998. Available at: <http://www-users.cs.umn.edu/~karypis/memis/hmetis/main.shtml>.
- [KK03] Victor N. Kravets and Prabhakar Kudva. Understanding metrics in logic synthesis for routability enhancement. In *Proceedings of the 2003 International Workshop on System Level Interconnect Prediction*, pages 3–6. ACM/SIGDA, 2003.
- [KM02] A. Kahng and S. Mantik. Measurement of inherent noise in EDA tools. In *Proceedings of the International Symposium on Quality in Electronic Design*, pages 206–211, 2002.
- [KO02] Kurt Keutzer and Michael Orshansky. From blind certainty to informed uncertainty. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 37–41. ACM/SIGDA, 2002.
- [KS01] Thomas Kutzschebauch and Leon Stok. Congestion aware layout driven logic synthesis. In *Proceedings of the 2001 International Conference on CAD*, pages 216–223, 2001.
- [KSD02] Prabhakar Kudva, Andrew Sullivan, and William Dougherty. Metrics for structural logic synthesis. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, 2002.
- [LKS01] Jinan Lou, Shankar Krishnamoorthy, and Henry S. Sheng. Estimating routing congestion using probabilis-

- tic analysis. In *Proceedings of the 2001 International Symposium on Physical Design*, pages 112–117. ACM/SIGDA, ACM Press, 2001.
- [LR71a] B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Trans. on Comput.*, C-20:1469–1479, 1971.
- [LR71b] B. S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Trans. on Comput.*, C-20:1469–1479, 1971.
- [LS91] Charles E. Leiserson and James B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6(1), 1991.
- [MB02a] Fan Mo and Robert K. Brayton. River PLA's: a regular circuit structure. In *Proceedings of the Design Automation Conference (DAC)*, pages 201–206, 2002.
- [MB02b] Fan Mo and Robert K. Brayton. Whirlpool PLA's: a regular logic structure and their synthesis. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, 2002.
- [MY87] A. Masaki and M. Yamada. Equations for estimating wire length in various types of 2-D and 3-D system packaging structures. *IEEE Trans. on computers, hybrids, and manufacturing Technology*, CHMT-10(2):190–198, June 1987.
- [NBHY89] Ravi Nair, C. Leonard Berman, Peter S. Hauge, and Ellen J. Yoffa. Generation of performance constraints for layout. *IEEE Transactions on Computer-Aided Design*, 8(8), 1989.
- [NR98] Sudip K. Nag and Rob A. Rutenbar. Performance-driven simultaneous placement and routing for FPGA's. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(6):499–518, 1998.
- [OK02] Michael Orshansky and Kurt Keutzer. A general probabilistic framework for worst case timing analysis. In *Proceedings of DAC*, pages 556–561, 2002.

- [OV89] R. H. J. M. Otten and L. P. P. P. Van Ginneken. *The Annealing Algorithm*. Kluwer Academic Publishers, 1989.
- [PMC⁺03] A. Papanikolaou, M. Mirande, F. Catthoor, H. Corporaal, H. De Man, D. De Roest, M. Stucchi, and K. Maex. Global interconnect trade-off for technology over memory modules to application level: Case study. In *Proceedings of the International Workshop on System-Level Interconnect Prediction*, pages 125–132. ACM/SIGDA, ACM Press, 2003.
- [RFCR99] A. Rahman, A. Fan, J. Chung, and R. Reif. Wire-length distribution of three-dimensional integrated circuits. *Proc. Int. Interconnect Technology Conf.*, pages 233–235, 1999.
- [Sak93] T. Sakurai. Closed-form expressions for interconnection delay, coupling and cross-talk in VLSI's. *IEEE Trans. on Electronic Devices*, 40:118–184, 1993.
- [SB02] Deshanand P. Singh and Stephen D. Brown. Incremental placement for layout-driven optimizations on FPGAs. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, 2002.
- [Sch02] Lou Scheffer. Explicit computation of performance as a function of process variation. In *Proceedings of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, pages 1–8. ACM/SIGDA, 2002.
- [She02] Naveed Sherwani. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, third edition, 2002.
- [SK99] D. Sylvester and K. Keutzer. System-level performance modeling with BACPAC - Berkeley Advanced Chip Performance Calculator. In *Proceedings of the IEEE/ACM International Workshop on System Level Interconnect Prediction*, pages 109–114. ACM, ACM Press, 1999.
- [SKT96] Majid Sarrafzadeh, David Knol, and Gustavo E. T  llez. A delay budgeting algorithm ensuring maximum flex-

- ibility in placement. In *Proceedings of the 1996 Physical Design Workshop*, 1996.
- [SLI99] 1st Workshop on System-Level Interconnect Prediction. <http://www.sliponline.org/>, 1999.
- [SM99] G. R. Schreiber and O. C. Martin. Cut size statistics of graph bisection heuristics. *SIAM Journal of Optimization*, 10(1):231–251, 1999.
- [SN00] Lou Scheffer and Eric Nequist. Why interconnect prediction doesn't work. In *Proceedings of the International Workshop on System Level Interconnect Prediction*, pages 139–144, 2000.
- [Sou81] J. Soukup. Circuit layout. *Proceedings of the IEEE*, pages 1281–1304, 1981.
- [SP86] S. Sastry and A. C. Parker. Stochastic models for wireability analysis of gate arrays. *IEEE Trans. Comput.-Aided Des., Integrated Circuits & Syst.*, 5:52–65, 1986.
- [SP91] Sarma Sastry and Jen-I Pi. Estimating the minimum of partitioning and floorplanning problems. *IEEE Trans. on Computer-Aided Design*, 10(2):273–282, 1991.
- [SSBK00] K.C. Saraswat, S.J. Souri, K. Banerjee, and P. Kapur. Performance analysis and technology of 3-D ICs. In *Proc. ACM Intl. Workshop on System-Level Interconnect Prediction*, pages 85–90. ACM Press, April 2000.
- [Str98] D. Stroobandt. Analytical methods for a priori wire length estimates in computer systems, November 1998. Ph.D. thesis (translated from Dutch), Ghent University, Engineering Faculty.
- [Str99] D. Stroobandt. On an efficient method for estimating the interconnection complexity of designs and on the existence of region III in Rent's rule. In M. A. Bayoumi and G. Jullien, editors, *Proc. 9th Great Lakes Symposium on VLSI*, pages 330–331. IEEE Computer Society Press, March 1999.

- [Str01a] D. Stroobandt. Multi-terminal nets do change conventional wire length distribution models. In *Proceedings of the 2001 International Workshop on System-Level Interconnect Prediction*, pages 41–48, April 2001.
- [Str01b] D. Stroobandt. *A priori Wire Length Estimates for Digital Design*. Kluwer Academic Publishers, April 2001.
- [SVC97] D. Stroobandt and J. Van Campenhout. Estimating interconnection lengths in three-dimensional computer systems. *IEICE Trans. on Inf. & Syst., Special Issue on Synthesis and Verification of Hardware Design*, E80–D(10):1024–1031, October 1997.
- [SVMVC97] D. Stroobandt, H. Van Marck, and J. Van Campenhout. Estimating logic cell to I/O pad lengths in computer systems. In T. Sasao, editor, *Proc. Workshop on Synthesis and System Integration of Mixed Technologies, SASIMI'97*, pages 192–198. S. Insatsu, Osaka, Japan, December 1997.
- [SVVC00] D. Stroobandt, P. Verplaetse, and J. Van Campenhout. Generating synthetic benchmark circuits for evaluating CAD tools. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(9):1011–1022, September 2000.
- [SWY03] M. Sarrafzadeh, M. Wang, and X. Yang. *Modern Placement Techniques*. Kluwer Academic Publishers, 2003.
- [UKK02] Junhyung Um, Jaehoon Kim, and Taewhan Kim. Layout-driven resource sharing in high-level synthesis. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, 2002.
- [VCVMDD99] J. Van Campenhout, H. Van Marck, J. Depreitere, and J. Dambre. Optoelectronic FPGAs. *IEEE Journal of Selected Topics in Quantum Electronics on Smart Photonic Components, Interconnects, and Processing*, 5(2):306–315, March/April 1999.
- [VDSVC01] P. Verplaetse, J. Dambre, D. Stroobandt, and J. Van Campenhout. On partitioning vs. placement Rent properties. In *Proc. of Intl. Workshop on*

- System-Level Interconnect Prediction*, pages 33–40, March 2001.
- [Ver03] P. Verplaetse. Karakterisatie van de interconnectie-topologie van digitale schakelingen en toepassingen in digitaal ontwerp, May 2003. Ph.D. thesis (in Dutch), Ghent University, Engineering Faculty.
- [VM00] H. Van Marck. Kwantitatieve studie en modellering van opto-elektronische driedimensionale interconnectiestructuren, March 2000. Ph.D. thesis (in Dutch), Ghent University, Engineering Faculty.
- [VMSVC95a] H. Van Marck, D. Stroobandt, and J. Van Campenhout. Interconnection length distributions in 3-dimensional anisotropic systems. In M. H. Hamza, editor, *Proc. 13th IASTED Intl. Conf. on APPLIED INFORMATICS*, pages 98–101. IASTED-ACTA Press, 1995.
- [VMSVC95b] H. Van Marck, D. Stroobandt, and J. Van Campenhout. Towards an extension of Rent’s rule for describing local variations in interconnection complexity. In S. Bai, J. Fan, and X. Li, editors, *Proc. 4th Intl. Conf. for Young Computer Scientists*, pages 136–141. Peking University Press, 1995.
- [WYES00] M. Wang, X. Yang, K. Eguro, and M. Sarrafzadeh. Multi-center congestion estimation and minimization during placement. In *Proceedings of the 2000 International Symposium on Physical Design*, pages 147–152. ACM/SIGDA, ACM Press, 2000.
- [YCS02] X. Yang, B-K. Choi, and M. Sarrafzadeh. Timing-driven placement using design hierarchy guided constraint generation. *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 2002.
- [YKS02] Xiaojian Yang, Ryan Kastner, and Majid Sarrafzadeh. Congestion estimation during top-down placement. *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 21(1):72–80, 2002.

- [YM03] Chao-Yang Yeh and Malgorzata Marek-Sadowska. Sequential delay budgeting with interconnect prediction. In *Proceedings of the 2003 International Workshop on System-level Interconnect prediction*, pages 23–30. ACM/SIGDA, 2003.
- [ZH01] Payman Zarkesh-Ha. Global interconnect modeling for a gigascale system-on-a-chip (GSoC), February 2001. Ph.D. thesis, Georgia Institute of Technology.
- [ZHDLM00] P. Zarkesh-Ha, J. A. Davis, W. Loh, and J. D. Meindl. Prediction of interconnect fan-out distribution using Rent’s rule. In *Proc. ACM Intl. Workshop on System-Level Interconnect Prediction*, pages 107–112. ACM Press, April 2000.